Eulerian Geometric Discretizations of Manifolds and Dynamics

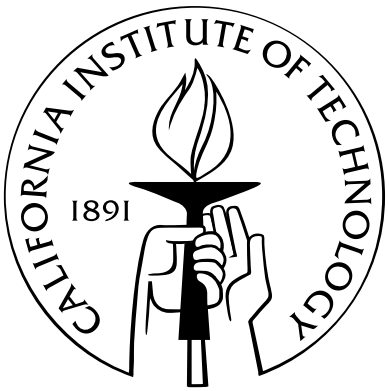
Thesis by

Patrick Mullen

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2012

(Defended September 8, 2011)

Acknowledgements

First I would like to thank my committee members—my advisor, Mathieu Desbrun, for both his mentorship and friendship, making graduate school an enriching and incredibly enjoyable experience; Peter Schröder, for first introducing me to research and continuing to give me numerous opportunities; Pierre Alliez, for several productive and enjoyable collaborations as well as the opportunity to work abroad; and Al Barr for his advice and long discussions, which always offered an enlightening perspective. I am very grateful to have had the opportunity to work with the late Jerrold E. Marsden, a kind and brilliant man who always left me contemplative and invigorated. I would like to thank both Houman Owhadi and David Cohen-Steiner for their mathematical consultations and advice, and Alain Martin for countless intriguing lunchtime discussions. I would also like to thank my many collaborators over the course of my graduate work, including Yiying Tong, Lily Kharevych, Dmitry Pavlov, Keenan Crane, Alexander McKenzie, Fernando de Goes, Pooran Memari, Evan Gawlik, and Luke Durant.

Finally, I would like to thank my parents and siblings for their constant love and support in whatever I chose to do, and my wife for her love, companionship, and for saying "I do."

Abstract

This thesis explores new methods for geometric, structure-preserving Eulerian discretizations of dynamics, including Lie advection and incompressible fluids, and the manifolds in which these dynamics occur. The result is a novel method for discrete Lie advection of differential forms, a new family of structure-preserving fluid integrators, and a new set of energies for optimizing meshes appropriate for some discrete geometric operators. First, high-resolution finite volume methods are leveraged to introduce a new method for discretizing the Lie advection of discrete differential forms, along with the related contraction operator, on regular grids. Through its geometric approach, the method exactly preserves properties such as the closedness of Lie advected closed forms. This results in an extension of finite volume techniques applicable to forms of arbitrary degree. After this, attention is turned to simplicial meshes, where new meshing techniques are developed to give formal error bounds on the discrete diagonal Hodge star, an important operator for geometric computations. Utilizing weighted Delaunay triangulations, both the primal mesh and its dual are optimized simultaneously over the entire space of orthogonal primal/dual pairs. Improved accuracy of the solution of Poisson equations is demonstrated as a practical application, as well as an increase in percentage of well-centered elements. Finally, a new structure-preserving method for the incompressible Navier-Stokes equations on simplicial meshes is developed, offering in the inviscid case the exact conservation of either the discrete energy or symplectic form. This leads to capturing the correct energy decay when viscosity is added, resulting in dissipation independent of grid and time resolution.

Contents

List of Figures

Chapter 1

Introduction

> "I'm not really a fan of using quotations, you usually just end up regretting it."—*Mathieu Desbrun*

Geometric discretizations and methods are a powerful tool gaining more and more attention in a wide spectrum of computation environments, ranging from geometry processing to Hamiltonian systems. By respecting symmetries of their continuous counterparts, these approaches have resulted in a variety of desirable properties, often in practice surprisingly outperforming their known theoretical guarantees. The purpose of this thesis is to continue developing the state of the art of such methods, in particular in the setting of Eulerian discretizations where much less work has been done, and to present some applications in computer graphics. However, in order to do so we must first review the current developments and terminology.

1.1 Setting the Stage

1.1.1 Structure Preservation: Identities, Invariants, and Symmetries

The notion of structure preservation will be pervasive in this thesis, and the term is used to cover a range of ideas. In its simplest form it can refer to the preservation of identities in the continuous domain. For example, the discrete exterior derivative, discussed in section 1.2, satisfies the identity $\mathbf{dd} = 0$, a fundamental property of the continuous exterior derivative encompassing several basic vector calculus identities. Preserving identities in the discrete world allows proofs built on these identities to carry over naturally to the discrete world as well, giving rise to additional properties preserved by the discrete system, such as existence and uniqueness of solutions to Poisson problems. Structure preservation is also used to describe the preservation of invariants of dynamic systems. For example, rigid body dynamics exactly preserve linear momentum, and hence it would be desirable for a rigid body simulator to also preserve a discrete notion of linear momentum, reducing the likelihood of erratic or completely unphysical behavior. Such invariants often arise from a symmetry of the

action of the physical system in the continuous world through Noether's theorem, which connects symmetries of the action to corresponding conserved quantities. For this reason, preserving the symmetries of systems is deeply connected with preserving invariants, and frequently the term symmetry is used almost interchangeably with invariant. Finally, the study of symmetries and invariants is a, if not *the*, key purpose of geometry, thanks to the body of work dating back to Felix Klein and his Erlangen Program. Hence, the term "geometric methods" is often used to refer to methods designed to preserve symmetries and invariants, and geometry plays an important role in their development.

1.1.2 Lagrangian vs Eulerian

When discretizing space for the purposes of physical simulations approaches often fall into two categories. Methods are referred to as *Lagrangian* when the location of the degrees of freedom (DOFs) have fixed material coordinates—i.e. they are "stuck" to the medium being simulated and move in the embedding space along with the material. Examples of Lagrangian discretizations include particles representing fluid molecules [1] or a mesh representing the surface of some deformable object [2]. The particles move around in the embedding space to represent the motion of the fluid, and similarly the mesh vertices move as the object deforms. At the other end of the spectrum are *Eulerian* methods, where now the DOFs have fixed spatial coordinates in the embedding space, and the simulated material can move with respect to them. Common examples include level set methods [3, 4], finite volumes [5], and most methods employing a fixed "background" grid for computation. For instance, in a finite volume method one may store the total volume of fluid in each cell of a fixed grid. As the fluid moves, the volume for each cell is updated, but the location of the cells remain unchanged.

Both Lagrangian and Eulerian methods have their advantages and disadvantages. Accurate tracking of moving surfaces and interfaces is often easier with Lagrangian methods since the DOFs can be kept exactly on the surface, while Eulerian methods often suffer from diffusion resulting from the fixed nature of the DOFs. On the other hand, many Lagrangian methods have difficulty dealing with topology changes, where Eulerian methods can handle these more naturally. Lagrangian methods can also have trouble converging to the proper vanishing viscosity solutions without extra care for shocks and rarefactions [4], and large deformations can lead to badly shaped or distributed elements causing other numerical issues. The handling of boundary conditions can vary greatly between the two, with Eulerian methods often proving convenient for dealing with complex fixed domain boundaries and Lagrangian methods excelling at accurate interface boundary conditions.

More advanced methods often combine or compromise between strict Lagrangian and Eulerian discretizations. Particle level sets [6] employ both a set of Lagrangian particles for better surface tracking along with an Eulerian grid for tracking topology changes. Arbitrary Lagrangian-Eulerian (ALE) methods [7] use meshes that deform based on the current simulation state without having

them exactly track the material, resulting in DOFs with both material and spatial coordinate varying. While this thesis will explore structure preservation on the strictly Eulerian side of the spectrum, extending these results to such hybrid methods is an area of active and interesting research.

1.1.3 Spatial Geometric Discretizations and DEC

With the goal of providing geometric approaches to PDE discretizations, structure-preserving geometric computational methods have emerged, gaining acceptance among engineers as well as mathematicians [8]. Computational electromagnetism [9, 10], mimetic (or natural) discretizations [11, 12], and more recently Discrete Exterior Calculus (DEC, [13, 14]) and Finite Element Exterior Calculus (FEEC, [15]) have all proposed similar discrete structures that discretely preserve vector calculus identities to obtain improved numerics. In particular, the relevance of exterior calculus (Cartan's calculus of differential forms [16]) and algebraic topology (see, for instance, [17]) to computations came to light.

Exterior calculus is a concise formalism to express differential and integral equations on smooth and curved spaces in a consistent manner, while revealing the geometrical invariants at play. At its root is the notion of differential forms, denoting antisymmetric tensors of arbitrary order. As integration of differential forms is an abstraction of the measurement process, this calculus of forms provides an intrinsic, coordinate-free approach particularly relevant to concisely describe a multitude of physical models that make heavy use of line, surface, and volume integrals [18, 19, 20, 21, 22, 23, 24]. Similarly, many physical measurements, such as fluxes, are performed as specific local integrations over a small surface of the measuring instrument. Pointwise evaluation of such quantities does not have physical meaning; instead, one should manipulate those quantities only as geometrically meaningful entities integrated over appropriate submanifolds—these entities and their geometric properties are embodied in discrete differential forms.

Algebraic topology, specifically the notion of chains and cochains (see, e.g., [25, 17]), has been used to provide a natural discretization of these differential forms and to emulate exterior calculus on finite grids: a set of values on vertices, edges, faces, and cells are proper discrete versions of respectively pointwise functions, line integrals, surface integrals, and volume integrals. This point of view is entirely compatible with the treatment of volume integrals in finite volume methods, or scalar functions in finite element methods [12]; but it also involves the edge elements and facet elements as introduced in E&M as special H_{div} and H_{curl} basis elements [26]. Equipped with such discrete forms of arbitrary degree, Stokes' theorem connecting differentiation and integration is automatically enforced if one thinks of differentiation as the dual of the boundary operator—a particularly simple operator on meshes. With these basic building blocks, important structures and invariants of the continuous setting directly carry over to the discrete world, culminating in a discrete Hodge theory (see recent progress in [27]). As a consequence, such a discrete exterior calculus has,

as we have mentioned, already proven useful in many areas such as electromagnetism [9, 10], fluid simulation [28], surface parameterization [29], and remeshing of surfaces [30] to mention a few.

1.1.4 Temporal Geometric Discretizations

Spatial discretizations are not the only components lending themselves to geometric methods. Many methods for numerical integration of ODEs and PDEs have been developed as, or later discovered to be, geometric methods with various structure-preserving properties. An excellent overview and introduction focusing on ODEs can be found in [31]. The benefits of discretely respecting different continuous symmetries can vary greatly depending on both the system being simulated and the targeted application. Examples of commonly desired structures to preserve include mass, volume, momentum, and energy. One less obvious but increasingly common structure whose preservation is desired for Hamiltonian systems is the symplectic form. Integrators for Hamiltonian systems that preserve a discrete notion of the symplectic form are referred to as *symplectic integrators*, or also *variational integrators* as they can be derived through a variational principle. These integrators tend to perform very well in a wide range of applications, with backwards error analysis combined with some perturbation theory explaining some of these nice behaviors, while in practice they often behave well even outside of the domain of their theoretical guarantees. A similar phenomenon arises with *time-symmetric integrators* (often just shortened to symmetric integrators) when applied to reversible systems. Again, while backwards error analysis points to some of the nice behavior, in practice these integrators often outperform the expectations.

1.2 DEC Overview

Guided by Cartan's exterior calculus of differential forms on smooth manifolds, DEC offers a calculus on discrete manifolds that maintains the covariant nature of the quantities involved. Since the methods and operators of DEC are used extensively throughout this thesis, we will briefly review the relevant portions here. In this section continuous quantities and operators are distinguished from their discrete counterparts through a **bold** typeface We will assume our mesh, be it a regular grid or simplicial complex, forms an orientable 3-manifold cell complex $K = (V, E, F, C)$ with vertex set $V = \{v_i\}$, edge set $E = \{e_{ij}\}$, as well as face set F and cell set C. Each face and edge is assigned an arbitrary yet fixed intrinsic orientation, while vertices and cells always have a positive orientation. By convention, if a particular edge e_{ij} is positively oriented, then e_{ji} refers to the same edge with negative orientation, and similar rules apply for higher-dimensional mesh elements given even vs odd permutations of their vertex indexing. We will also need the notion of the dual mesh, constructed by assigning to each k-cell an $(n-k)$-cell, with connectivity and orientation given from the corresponding primal mesh. While the choice of an appropriate dual mesh for simplicial meshes

is the topic of chapter 3, we will generally restrict our discussion to orthogonal dual meshes, where the dual elements span the orthogonal space of their corresponding primal elements (see section 3.2 for more details). For a mesh element σ (be it primal or dual), we will use the notation $*\sigma$ to denote the element dual to that element.

Boundary operators. Assuming that mesh elements in K are enumerated with an arbitrary (but fixed) indexing, the incidence matrices of K then define the boundary operators. For example, we let ∂^1 denote the $|V| \times |E|$ matrix with $(\partial^1)_{ve} = 1$ (resp., -1) if vertex v is incident to edge e and the edge orientation points towards (resp., away from) v, and zero otherwise. Similarly, ∂^2 denotes the $|E| \times |F|$ incidence matrix of edges to faces with $(\partial^1)_{ef} = 1$ (resp., -1) if edge e is incident to face f and their orientations agree (resp., disagree), and zero otherwise. The incidence matrix of faces to cells ∂^3 is defined in a similar way. See [17] for details.

Chains and cochains. At the core of this computational tool is the notion of *chains*, defined as a linear combination of mesh elements; a 0-chain is a weighted sum of vertices, a 1-chain is a weighted sum of edges, etc. Since each k-dimensional cell has a well-defined notion of boundary (in fact its boundary is a chain itself; the boundary of a face, for example, is the signed sum of its edges), the boundary operator naturally extends to chains by linearity. A *discrete form* is simply defined as the dual of a chain, or *cochain*, a linear mapping that assigns each chain a real number. Thus, a 0-cochain (that we will abusively call a 0-form sometimes) amounts to one value per 0-dimensional cell, such that any 0-chain can naturally pair with this cochain. More generally, k-cochains are defined by one value per k-cell, and they naturally pair with k-chains. The resulting pairing of a k-cochain α^k and a k-chain σ_k is the discrete equivalent of the integration of a continuous k-form $\boldsymbol{\alpha}^k$ over a k-dimensional submanifold $\boldsymbol{\sigma}_k$:

$$\int_{\boldsymbol{\sigma}_k} \boldsymbol{\alpha}^k \equiv \langle \alpha^k, \sigma_k \rangle.$$

Alternatively, chains can be defined on the dual mesh, and the same ideas carry over. For example, where as a primal discrete 0-form assigns each vertex of the primal mesh a real number, we will refer to a dual 0-form as the discrete form that assigns each dual vertex a real number. While attractive from a computational perspective due to their conceptual simplicity and elegance, the chain and cochain representations are also deeply rooted in a theoretical framework defined by H. Whitney [25], who introduced the Whitney and deRham maps that establish an isomorphism between the cohomology of simplicial cochains and the cohomology of Lipschitz differential forms. With these theoretical foundations, chains and cochains are used as basic building blocks for direct discretizations of important geometric structures such as the deRham complex through the introduction of two simple operators: the exterior derivative and the Hodge star.

Exterior derivative. The differential $\mathbf{d}$ (called exterior derivative) is an existing exterior calculus operator that we will need in our construction of a Lie derivative. The discrete derivative d is constructed to satisfy Stokes' theorem, which elucidates the duality between the exterior derivative and the boundary operator. In the continuous sense, it is written

$$\int_{\boldsymbol{\sigma}} \mathbf{d}\boldsymbol{\alpha} = \int_{\boldsymbol{\partial\sigma}} \boldsymbol{\alpha}. \tag{1.1}$$

Consequently, if α is a discrete differential k-form, then the $(k{+}1)$-form $\mathrm{d}\alpha$ is defined on any $(k{+}1)$-chain σ by

$$\langle \mathrm{d}\alpha, \sigma \rangle = \langle \alpha, \partial\sigma \rangle \,, \tag{1.2}$$

where $\partial\sigma$ is the (k-chain) boundary of σ, as defined in section 1.2. Thus the discrete differential d, mapping k-forms to $(k{+}1)$-forms, is given by the coboundary operator, the transpose of the signed incidence matrices of the complex K; $\mathrm{d}_0 = (\partial^1)^T$ maps 0-forms to 1-forms, $\mathrm{d}_1 = (\partial^2)^T$ maps 1-forms to 2-forms, and more generally in nD, $\mathrm{d}_k = (\partial^{k+1})^T$. In relation to standard 3-dimensional vector calculus, this can be seen as $\mathbf{d}_0 \equiv \boldsymbol{\nabla}$, $\mathbf{d}_1 \equiv \boldsymbol{\nabla}\times$, and $\mathbf{d}_2 \equiv \boldsymbol{\nabla}\cdot$. The fact that the boundary of a boundary is empty results in $\mathrm{dd} = 0$, which in turn corresponds to the vector calculus facts that $\boldsymbol{\nabla} \times \boldsymbol{\nabla} = \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\times = 0$. Notice that this operator is defined purely combinatorially, and thanks to the integrated nature of discrete forms combined with Stokes' Theorem is exact, and thus does not introduce error or need a high-order definition, unlike the remaining operators we introduce next.

Hodge star. The continuous Hodge star operator $\star$ arises in a variety of practical contexts, the most common being in the computation of the ubiquitous Laplacian operator, written for differential forms as $\star\mathbf{d}\star\mathbf{d}+\mathbf{d}\star\mathbf{d}\star$. In n dimensions the Hodge star maps a k-form to an $(n–k)$-form, such that at every point the new form has the same magnitude and spans the space orthogonal to that spanned by the original (or more formally, the wedge product of the two normalized forms is the volume form). Its simplest discretization, known as the diagonal Hodge star, utilizes this orthogonality along with that between the primal and dual mesh, hence mapping a primal k-form to a dual $(n-k)$-form and vice versa. This is done independently per primal/dual element pair, resulting in a diagonal matrix from which it derives its name. Its computation is based on the assumption that the form is locally constant, and therefore for a k-form α^k the integral of $\star\alpha^k$ over the dual element $*\sigma_i$ can be approximated by integrating the constant density $\frac{\alpha_i^k}{|\sigma_i|}$ over the orthogonal dual element. While this approximation is very low order and can introduce significant error, its computational simplicity and efficiency make it very appealing. Chapter 3 will discuss a new method to design meshes to minimize the price paid for this convenience. Note that in some cases we will use the notation $\star_k$ to refer to the particular Hodge star that is applied to primal k-forms, while in many cases the choice of star is clear from context and no subscript is used.

Lie derivative and contraction. The Lie derivative is another important operator in exterior calculus that generalizes the derivative of a function along a vector field to differential forms. Its discretization, along with that of the related contraction operator, are the topic of chapter 2 and will be discussed in detail there.

1.3 Outline

The remaining text of thesis will first discuss a novel approach on regular grids of Lie advection of differential forms of arbitrary degree. After this, simplicial meshes will be discussed, starting with an analysis of what properties of a mesh are attractive in DEC-like discretizations. This will be followed by an example application, where new geometric Navier-Stokes integrators are developed for Eulerian fluid simulation, including both energy-preserving and symplectic varieties. Finally, the important lessons learned over the course of this work are discussed, along with the abundant amount of potential future work it suggests.

Chapter 2

Discrete Lie Advection of Differential Forms

2.1 Introduction

In this chapter we introduce a finite volume based technique for solving the discrete Lie advection equation, ubiquitous in most advection phenomena:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\mathcal{L}}_{\boldsymbol{X}} \boldsymbol{\omega} = 0, \tag{2.1}$$

where $\boldsymbol{\omega}$ is an arbitrary discrete differential k-form [15, 12, 14] defined on a discrete manifold, and $\boldsymbol{X}$ is a discrete vector field living on this manifold. Our numerical approach stems from the observation, developed in this chapter, that the computational treatment of discrete differential forms share striking similarities with finite volume techniques [32] and scalar advection techniques used in level sets [3, 4]. Consequently, we present a discrete interior product (or *contraction*) computed using any of the k-dimensional finite volume methods readily available, from which we derive a numerical approximation of the spatial Lie derivative $\boldsymbol{\mathcal{L}}_{\boldsymbol{X}}$ using a combinatorial exterior derivative.

2.1.1 Background on the Lie Derivative

The notion of Lie derivative $\boldsymbol{\mathcal{L}}_{\boldsymbol{X}}$ in Elie Cartan's Exterior Calculus [16] extends the usual concept of the derivative of a function along a vector field $\boldsymbol{X}$. Although a formal definition of this operator can be made purely algebraically (see [19], section 5.3), its nature is better elucidated from a dynamical perspective [19] (section 5.4). Consequently, the spatial Lie derivative (along with its closely related time-dependent version) is an underlying element in all areas of mechanics: for example, the rate of strain tensor in elasticity and the vorticity advection equation in fluid dynamics are both nicely described using Lie derivatives.

A common context where a Lie derivative is used to describe a physical evolution is in the

advection of *scalar fields*: a scalar field $\boldsymbol{\rho}$ being advected in a vector field $\boldsymbol{V}$ can be written as: $\boldsymbol{\partial\rho/\partial t} + \boldsymbol{\mathcal{L}_V \rho} = 0$. The case of divergence-free vector fields (i.e., $\boldsymbol{\nabla\cdot V} = 0$) has been the subject of extensive investigation over the past several decades leading to several numerical schemes for solving these types of hyperbolic conservation laws in various applications (see, e.g., [33, 34, 35, 36, 37, 28, 38]). Chief among them are the so-called finite volume methods [32], including upwind, ENO, WENO, and high-resolution techniques. Unlike finite difference techniques based on point values (e.g., [39, 40, 41]), such methods often resort to the conservative form of the advection equation and rely on cell averages and the integrated fluxes in between. The integral nature of these finite volume techniques will be particularly suitable in our context, as it matches the foundations behind discrete versions of exterior calculus [12, 15].

While finite volume schemes have been successfully used for over a decade, they have been used almost solely to advect scalar fields, be they functions or densities, or systems thereof (e.g., components of tensor fields). To our knowledge, Lie advection of inherently nonscalar entities such as vorticity for fluids has yet to benefit from these advances, as such differential forms are not Lie advected in the same manner as scalar fields. Concurrent to the development of high-resolution methods for scalar advection, structure-preserving geometric computational methods have been emerging as well, as discussed in section 1.1.3. Despite this previous work, the contraction and Lie derivative of arbitrary discrete forms—two important operators in exterior calculus—have received very little attention, with a few exceptions. The approach in [42] (which we will review in section 2.3.1) is to exploit the duality between the extrusion and contraction operators, resulting in an integral definition of the interior product that fits the existing foundations. While a discrete contraction was derived using linear "Whitney" elements, no method to achieve low numerical diffusion and/or high resolution was proposed. Furthermore, the Lie derivative was not discussed. More recently Heumann and Hiptmair [43] leveraged this work to suggest an approach similar to ours in a finite element framework for Lie advection of forms of arbitrary degree, however only 0-forms were analyzed.

2.2 Mathematical Tools

Before introducing our contribution, we briefly review the existing mathematical tools we will need in order to derive a discrete Lie advection. We assume a regular Cartesian grid discretization of space, and will leverage the notation and DEC operators introduced in section 1.2. Given the integral representation of discrete forms, a last numerical tool we will need is a method for computing solutions to advection problems in integral form. Finite volume methods were developed for exactly this purpose, and while we now provide a brief overview of this general procedure for completeness, we refer the reader to [32] and references therein for further details and applications. One approach

of finite volume schemes is to advect a function $u(x)$ by a velocity field $v(x)$ using a Reconstruct-Evolve-Average (REA) approach. In one dimension, we can define the cell average of a function $u(x)$ over cell C_i with width Δx as

$$\bar{u}_i = \frac{1}{\Delta x} \int_{C_i} u(x)\, dx \quad i = 1, 2, \ldots, N.$$

Given k adjacent cell averages, the method will reconstruct a function such that the average of $p(x)$ in each of the k cells is equal to the average of $u(x)$ in those cells. High-resolution methods attempt to build a reconstruction such that it has only high-order error terms in smooth regions, while lowering the order of the reconstruction in favor of avoiding oscillations in regions with discontinuities like shocks. Such adaptation can be done through the use of slope limiters or by changing stencil sizes using essentially nonoscillatory (ENO) and related methods. This reconstruction can then be evolved by the velocity field and averaged back onto the Eulerian grid.

Another variant of finite volume methods is one that computes fluxes through cell boundaries. Employing Stokes' theorem, the REA approach can be implemented by computing only the integral of the reconstruction that is evolved through each face, and then differencing the incoming and outgoing integrated fluxes of each cell to determine its net change in density. It is this flux differencing approach that will be most convenient for deriving our discrete contraction operator, due to the observation that the net flux through a face induced by evolving a function forward in a velocity field is equal to the flux through the face induced by evolving the face *backwards* through the same velocity field. This second interpretation of the integrated flux is the same as computing the integral of the function over an extrusion of the face in the velocity field, as will be seen in the next section, and therefore we may use any of the wide range of finite volume methods to approximate integrals over extruded faces.

2.3 Discrete Interior Product and Discrete Lie Derivative

In keeping with the foundations of Discrete Exterior Calculus, we present the continuous interior product and Lie derivative operators in their "integral" form, i.e., we present continuous definitions of $\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega}$ and $\boldsymbol{\mathcal{L}}_{\boldsymbol{X}}\boldsymbol{\omega}$ *integrated* over infinitesimal submanifolds: these integral forms will be particularly amenable to discretization via finite volume methods and DEC as we discussed earlier.

2.3.1 Toward a Dynamic Definition of Lie Derivative

Interior product through extrusion. As pointed out by [42], the *extrusion* of objects under the flow of a vector field can be used to give an intuitive dynamic definition of the interior product. If $\boldsymbol{\mathcal{M}}$ is an n-dimensional smooth manifold and $\boldsymbol{X} \in \mathfrak{X}(\boldsymbol{\mathcal{M}})$ a smooth (tangent) vector field on the

manifold, let $\boldsymbol{\mathcal{S}}$ be a k-dimensional submanifold on $\boldsymbol{\mathcal{M}}$ with $k < n$. The flow $\boldsymbol{\varphi}$ of the vector field $\boldsymbol{X}$ is simply a function $\boldsymbol{\varphi} : \boldsymbol{\mathcal{M}} \times \mathbb{R} \to \boldsymbol{\mathcal{M}}$ consistent with the one-parameter (time) group structure, that is, such that $\boldsymbol{\varphi}(\boldsymbol{\varphi}(\boldsymbol{\mathcal{S}}, t), s) = \boldsymbol{\varphi}(\boldsymbol{\mathcal{S}}, s + t)$ with $\boldsymbol{\varphi}(\boldsymbol{\mathcal{S}}, 0) = \boldsymbol{\mathcal{S}}$ for all $s, t \in \mathbb{R}$. Now imagine that $\boldsymbol{\mathcal{S}}$ is carried by this flow of $\boldsymbol{X}$ for a time t; we denote the resultant "flowed-out" submanifold $\boldsymbol{\mathcal{S}}_{\boldsymbol{X}}(t)$, which is equivalent to the image of $\boldsymbol{\mathcal{S}}$ under the mapping $\boldsymbol{\varphi}$, i.e., $\boldsymbol{\mathcal{S}}_{\boldsymbol{X}}(t) \equiv \boldsymbol{\varphi}(\boldsymbol{\mathcal{S}}, t)$. The extrusion $\boldsymbol{E}_{\boldsymbol{X}}(\boldsymbol{\mathcal{S}}, t)$ is then the $(k{+}1)$-dimensional submanifold formed by the advection of $\boldsymbol{\mathcal{S}}$ over the time t to its final position $\boldsymbol{\mathcal{S}}_{\boldsymbol{X}}(t)$: it is the "extruded" (or "swept out") submanifold. This can be expressed formally as a union of flowed-out manifolds,

$$\boldsymbol{E}_{\boldsymbol{X}}(\boldsymbol{\mathcal{S}}, t) = \bigcup_{\tau \in [0,t]} \boldsymbol{\mathcal{S}}_{\boldsymbol{X}}(\tau),$$

where the orientation of $\boldsymbol{E}_{\boldsymbol{X}}(\boldsymbol{\mathcal{S}}, t)$ is defined such that

$$\partial \boldsymbol{E}_{\boldsymbol{X}} = \boldsymbol{\mathcal{S}}_{\boldsymbol{X}}(t) - \boldsymbol{\mathcal{S}} - \boldsymbol{E}_{\boldsymbol{X}}(\partial \boldsymbol{\mathcal{S}}, t).$$

These geometric notions are visualized in figure 2.1, where the submanifold $\boldsymbol{\mathcal{S}}$ is presented as a 1-dimensional curve, flowed out to form $\boldsymbol{\mathcal{S}}_{\boldsymbol{X}}(t)$, or alternatively, extruded to form $\boldsymbol{E}_{\boldsymbol{X}}(\boldsymbol{\mathcal{S}}, t)$.

Using this setup, the interior product $\mathbf{i}_{\boldsymbol{X}}$ of a time-independent form $\boldsymbol{\omega}$ evaluated on $\boldsymbol{\mathcal{S}}$ can now be defined through one of its most crucial properties, i.e., as the instantaneous change of $\boldsymbol{\omega}$ evaluated on $\boldsymbol{E}_{\boldsymbol{X}}(\boldsymbol{\mathcal{S}}, t)$, or more formally,

$$\int_{\boldsymbol{\mathcal{S}}} \mathbf{i}_{\boldsymbol{X}} \boldsymbol{\omega} = \left. \frac{d}{dt} \right|_{t=0} \int_{\boldsymbol{E}_{\boldsymbol{X}}(\boldsymbol{\mathcal{S}}, t)} \boldsymbol{\omega}. \tag{2.2}$$

While this equation is coherent with the discrete spatial picture, for the discrete Lie advection we will also wish to integrate $\mathbf{i}_{\boldsymbol{X}} \boldsymbol{\omega}$ over a small time interval. Hence, by taking the integral of both sides of equation (2.2) over the interval $[0, \Delta t]$, the first fundamental theorem of calculus gives us

$$\int_0^{\Delta t} \left[\int_{\boldsymbol{\mathcal{S}}_{\boldsymbol{X}}(t)} \mathbf{i}_{\boldsymbol{X}} \boldsymbol{\omega} \right] dt = \int_{\boldsymbol{E}_{\boldsymbol{X}}(\boldsymbol{\mathcal{S}}, \Delta t)} \boldsymbol{\omega}, \tag{2.3}$$

which will be used later on for the discretization of the time-integrated interior product.

Algebraic and flowed-out Lie derivative. Using a similar setup, we can formulate a definition of Lie derivative based on the flowed-out submanifold $\boldsymbol{\mathcal{S}}_{\boldsymbol{X}}(t)$. Remember that the Lie derivative is a generalization of the directional derivative to tensors, intuitively describing the change of $\boldsymbol{\omega}$ in the direction of $\boldsymbol{X}$. In fact, the Lie derivative $\boldsymbol{\mathcal{L}}_{\boldsymbol{X}} \boldsymbol{\omega}$ evaluated on $\boldsymbol{\mathcal{S}}$ is equivalent to the instantaneous

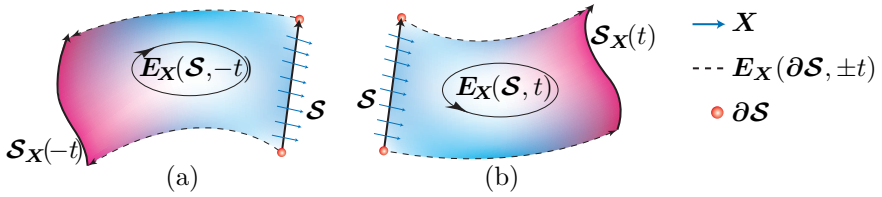


Figure 2.1. ***Geometric interpretation of the Lie derivative*** $\mathcal{L}_X\omega$ *of a differential form* ω *in the direction of vector field* X*: (a) for a backwards advection in time of an edge* $\mathcal{S}$ *(referred to as upwind extrusion), and (b) for a forward advection of* $\mathcal{S}$*. Notice the orientation of the two extrusions are opposite, and depend on the direction of the velocity field.*

change of ω evaluated on $\mathcal{S}_X(t)$, formally expressed by

$$\int_{\mathcal{S}} \mathcal{L}_X\omega = \frac{d}{dt}\bigg|_{t=0} \int_{\mathcal{S}_X(t)} \omega, \tag{2.4}$$

as a direct consequence of the Lie derivative theorem [19](Theorem 6.4.1). As before, we can integrate equation (2.4) over a small time interval $[0, \Delta t]$, applying the Newton-Leibnitz formula to find

$$\int_0^{\Delta t} \left[\int_{\mathcal{S}_X(t)} \mathcal{L}_X\omega \right] dt = \int_{\mathcal{S}_X(\Delta t)} \omega - \int_{\mathcal{S}} \omega. \tag{2.5}$$

Note that the formulation above, discretized using a semi-Lagrangian method, has been used, e.g., by [28] to advect fluid vorticity; in that case the right-hand side of equation (2.5) was evaluated by looking at the circulation through the boundary of the "backtracked" manifold. Rather than following their approach, we revert to discretizing the dynamic definition of the interior product in equation (2.3) instead, and later constructing the Lie derivative algebraically. The primary motivation behind this modification is one of effective numerical implementation: we can apply a dimension-by-dimension finite volume scheme to obtain an approximation of the interior product, while the alternative—computing integrals of approximated ω over $\mathcal{S}_X(t)$ as required by a discrete version of equation (2.5)—is comparatively cumbersome. Also, by building on top of standard finite volume schemes the solvers can leverage pre-existing code, such as CLAWPACK [44], without requiring modification.

We now show how the Lie derivative and the interior product are linked through a simple algebraic relation known as Cartan's homotopy formula. In particular, this derivation (using figure 2.1 as a reference) requires repeated application of Stokes' theorem from equation (1.1).

$$\lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_0^{\Delta t} \left[\int_{\mathcal{S}_X(t)} \mathcal{L}_X\omega \right] dt = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left[\int_{\mathcal{S}_X(\Delta t)} \omega - \int_{\mathcal{S}} \omega \right] \tag{2.6}$$

$$= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left[\int_{\boldsymbol{E_X}(\boldsymbol{\mathcal{S}}, \Delta t)} \mathbf{d}\boldsymbol{\omega} + \int_{\boldsymbol{E_X}(\boldsymbol{\partial \mathcal{S}}, \Delta t)} \boldsymbol{\omega} \right] \tag{2.7}$$

$$= \int_{\boldsymbol{\mathcal{S}}} \mathbf{i}_{\boldsymbol{X}} \mathbf{d}\boldsymbol{\omega} + \int_{\boldsymbol{\partial \mathcal{S}}} \mathbf{i}_{\boldsymbol{X}} \boldsymbol{\omega} \tag{2.8}$$

$$= \int_{\boldsymbol{\mathcal{S}}} \mathbf{i}_{\boldsymbol{X}} \mathbf{d}\boldsymbol{\omega} + \int_{\boldsymbol{\mathcal{S}}} \mathbf{d}\mathbf{i}_{\boldsymbol{X}} \boldsymbol{\omega}. \tag{2.9}$$

The submanifolds $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{S}_X}(\Delta t)$ form a portion of the boundary of $\boldsymbol{E_X}(\boldsymbol{\mathcal{S}}, \Delta t)$. Therefore, by Stokes', we can evaluate $\mathbf{d}\boldsymbol{\omega}$ on the extrusion and subtract off the other portions of $\boldsymbol{\partial E_X}(\boldsymbol{\mathcal{S}}, \Delta t)$ to obtain the desired quantity. This is how we proceed from equation (2.6) to equation (2.7) of the proof. The following line, equation (2.8), is obtained by applying the dynamic definition of the interior product given in equation (2.3) to each of two terms, leading us to our final result in equation (2.9) through one final application of Stokes' theorem. What we have obtained is the Lie derivative expressed algebraically in terms of the exterior derivative and interior product. Notice that equation (2.9) is the integral form of the celebrated identity called Cartan's homotopy (or *magic*) formula, most frequently written as

$$\boldsymbol{\mathcal{L}_X} \boldsymbol{\omega} = \mathbf{i}_{\boldsymbol{X}} \mathbf{d}\boldsymbol{\omega} + \mathbf{d}\mathbf{i}_{\boldsymbol{X}} \boldsymbol{\omega}. \tag{2.10}$$

By defining our discrete Lie derivative through this relation, we ensure the algebraic definition holds true in the discrete sense by construction. It also implies that the Lie derivative can be directly defined through interior product and exterior derivative, without the need for its own discrete definition.

Upwinding the extrusion. We may rewrite the above notions using an "upwinded" extrusion (i.e., a cell extruded backwards in time) as well (see figure 2.1a). For example, equation (2.2) can be rewritten as

$$\int_{\boldsymbol{\mathcal{S}}} \mathbf{i}_{\boldsymbol{X}} \boldsymbol{\omega} = -\frac{d}{dt}\bigg|_{t=0} \int_{\boldsymbol{E_X}(\boldsymbol{\mathcal{S}}, -t)} \boldsymbol{\omega}. \tag{2.11}$$

While this does not change the instantaneous value of the contraction, integrating equation (2.11) over the time interval $[0, \Delta t]$ now gives us

$$\int_0^{\Delta t} \left[\int_{\boldsymbol{\mathcal{S}_X}(t)} \mathbf{i}_{\boldsymbol{X}} \boldsymbol{\omega} \right] dt = - \int_{\boldsymbol{E_X}(\boldsymbol{\mathcal{S}}, -\Delta t)} \boldsymbol{\omega}. \tag{2.12}$$

Similar treatment for the remainder of the above can be done and Cartan's formula can be derived the same way, however by using these definitions in our following discretization we will obtain computations over upwinded regions equivalent to those computed by finite volume methods.

2.3.2 Discrete Interior Product

A discrete interior product is computed by exploiting the principles of equation (2.3) and applying the finite volume machinery. Given a discrete k-form α and a discrete vector field X, the interior product is approximated by extruding backwards in time every (k-1)-dimensional cell σ of the computational domain to form a new k-dimensional cell $E_X(\sigma, -\Delta t)$. Evaluating the integral of α over the extrusion and assigning the resulting value to the original cell σ yields the mapping $\langle \mathrm{i}_X \alpha, \sigma \rangle$ integrated over a time step Δt. This procedure, once applied to all (k-1)-dimensional cells, gives the desired discrete (k-1)-form $\mathrm{i}_X \alpha$.

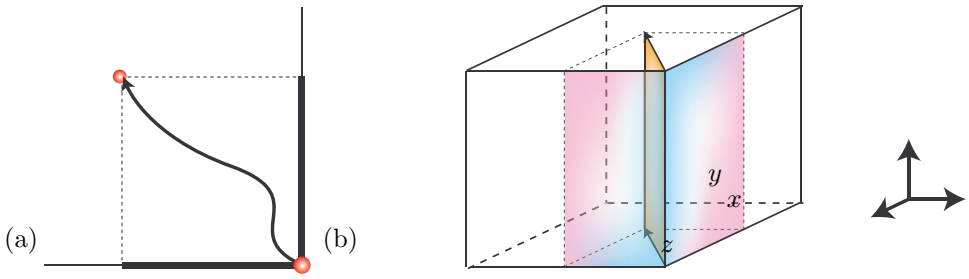


Figure 2.2. ***Approximating Extrusions****: In the discrete setting, the extrusion of a (k-1)-dimensional manifold (k=1 on left, 2 on right) is approximated by projecting the Lagrangian advection of the manifold into $\binom{n}{k}$ separate k-dimensional components.*

K-dimensional splitting. One option for computing this integral would be to do an n-dimensional reconstruction of α, perform a Lagrangian advection of the cell σ to determine $E_X(\sigma, -\Delta t)$, and then algebraically or numerically computing the integral of the reconstructed α over this extrusion. In fact, this is the idea behind the approach suggested in [43]. However, with the exception of when $k = n$, such an approach does not allow us to directly leverage finite volume methods, as performing an n-dimensional reconstruction of a form given only integrals over k-dimensional submanifolds would require a more general finite element framework. For simplicity and ease of implementation we avoid this generalization and instead resort to projecting the extrusion onto the grid-aligned k-dimensional subspaces and then applying a k-dimensional finite volume method to each of the $\binom{n}{k}$ projections. The integrals over the extrusion of σ from each dimension are then summed. Again, note that in the special case of $k = n$ no projection is required and we are left exactly with an n-dimensional finite volume scheme. We have found that this splitting combined with a high-resolution finite volume method, despite imposing at most first order accuracy, can still give high quality results with low numerical diffusion, while being able to leverage existing finite volume solvers without modification. However, if truly higher order is required then a full-blown finite element method would most likely be required [43].

Finite volume evaluation. As hinted at in section 2.2, we notice that the time integral of the flux of a density field being advected through a submanifold σ is equivalent to the integral of the density field over the backwards extrusion of σ over the same amount of time. In fact, some finite volume methods are derived using this interpretation, doing a reconstruction of the density field, approximating the extrusion, and integrating the reconstruction over this. However, many others are explained by computing a numerical flux per face, and then multiplying by the time step Δt: this is still an approximation of the integral over the extrusion, taking the reconstruction to be a constant (the numerical flux divided by v) and the extrusion having length $v\Delta t$. Indeed the right hand side of equation (2.2) can be seen as analogous to the numerical flux, after which equation (2.3) becomes the relationship between integrating the flux over time and the form over the extrusion. Hence we may use any of the finite volume methods for k-dimensional density advection problems when computing the contraction of a k-form. The only difference here is that rather than applying Stokes theorem and summing the contributions back to the original k-cell (which will be done by the discrete exterior derivative in the $\mathbf{di}_{\boldsymbol{X}}\boldsymbol{\omega}$ term of the Lie derivative), the contraction simply stores the values on the (k-1)-cells, without the final sum.

2.3.3 Discrete Lie Advection

We now have all the ingredients to introduce a discrete Lie advection. Given a k-form α, we compute the (k+1)-form $\mathrm{d}\alpha$ by applying the transpose of the incidence matrix ∂^{k+1} to α as detailed in section 1.2. We then compute the k-form $\mathrm{i}_X(\mathrm{d}\alpha)$, and the ($k$-1)-form $\mathrm{i}_X\alpha$. By applying d to the latter form and summing the resulting k-form with the other interior product, we finally get an approximation of Cartan's homotopy formula of the Lie derivative. An explicit example of this will be given in the next section to better illustrate the process and details.

2.4 Applications and Results

We now present a few direct applications of this discrete Lie advection scheme. In our tests we used upwinding one-dimensional WENO schemes for our contraction operator, splitting even the k-dimensional problems into multiple one-dimensional ones. We found that when using high-resolution WENO schemes we could obtain quality results with little numerical smearing despite this dimensional splitting.

A note on vector fields. In this section we assume that vector fields are discretized by storing their flux (i.e., contraction with the volume form) on all the ($n-1$)-dimensional cells of a nD regular grid, much like the Marker-And-Cell "staggered" grid setup [45]. Evaluation of the vector fields at lower dimensional cells is done through simple averaging of adjacent discrete fluxes. We pick this

setup as it is one of the most commonly-used representations, but the vector fields can be given in arbitrary form with only minor implementation changes.

2.4.1 Volume Forms and 0-Forms

Applying our approach to volume forms (n-forms in n dimensions) we have

$$\boldsymbol{\mathcal{L}}_{\boldsymbol{X}}\boldsymbol{\omega} = \mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega} + \mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega} = \mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega}.$$

Note that $\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega}$ is the numerical flux computed by the chosen n-dimensional finite volume scheme while $\mathbf{d}$ will then just assign the appropriate sign of this flux to each cell's update, and hence we trivially arrive at the chosen finite volume scheme with no modification. Similarly, applying this approach to 0-forms results in well-known finite difference advection schemes of scalar fields. Indeed, we have in this case

$$\boldsymbol{\mathcal{L}}_{\boldsymbol{X}}\boldsymbol{\omega} = \mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega} + \mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega} = \mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega},$$

as the contraction of a 0-form vanishes. We are thus left with $\mathbf{d}\boldsymbol{\omega}$ computing standard finite differences of a node-based scalar field on edges, and $\mathbf{i}_{\boldsymbol{X}}$ then doing componentwise upwind integration of reconstructions of these derivatives. Such techniques are common in scalar field advection, for example in the advection of level sets, and we refer the reader to [4, 3] and references therein for examples.

2.4.2 Advecting a 1-Form in 2 dimensions

The novelty of this approach comes when applied to k-forms in n dimensions with $n > k > 0$. We first demonstrate the simplest such application of our method by advecting a 1-form by a static velocity field in 2 dimensions using the simple piecewise-constant upwinding finite volume advection. To illustrate the general approach we will explicitly write out the algorithm for this case. We will assume the velocity $\boldsymbol{X}$ is everywhere positive in both x and y components to simplify the upwinding, and X^x and X^y will be used to represent the integrated flux through vertical and horizontal edges respectively.

Suppose we have a regular two-dimensional grid with square cells of size h^2, and with each horizontal edge oriented in the positive x direction and each vertical edge oriented in the positive y direction and numbered according to figure 2.3. A discrete 1-form $\boldsymbol{\omega}$ is represented by its integral along each edge. Due to the Cartesian nature of the grid, this implies that the dx component of the form will be stored on horizontal edges and the dy component will be stored on vertical edges, and we represent these scalars as $\boldsymbol{\omega}^x_{i,j}$ and $\boldsymbol{\omega}^y_{i,j}$ for the integrals along the (i, j) horizontal and vertical edge respectively. The discrete exterior derivative integrated over cell (i, j), $(\mathbf{d}\boldsymbol{\omega})_{i,j}$, consists of the

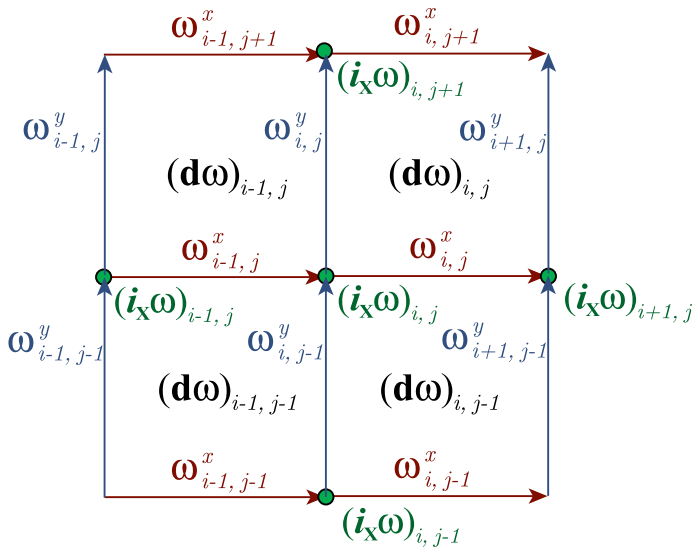


Figure 2.3. ***Grid setup:*** *Indexing and location of the various quantities stored on different parts of the grid. Arrows indicate the orientation of the edges.*

signed sum of $\boldsymbol{\omega}$ over cell (i, j)'s boundary edges, namely

$$(\mathbf{d}\boldsymbol{\omega})_{i,j} = \boldsymbol{\omega}^x_{i,j} + \boldsymbol{\omega}^y_{i+1,j} - \boldsymbol{\omega}^x_{i,j+1} - \boldsymbol{\omega}^y_{i,j}.$$

Using piecewise-constant upwind advection, and remembering the assumption of positivity of the components of $\boldsymbol{X}$, we may now compute $\mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega}$ over a time interval Δt for the horizontal and vertical edges (i, j) as

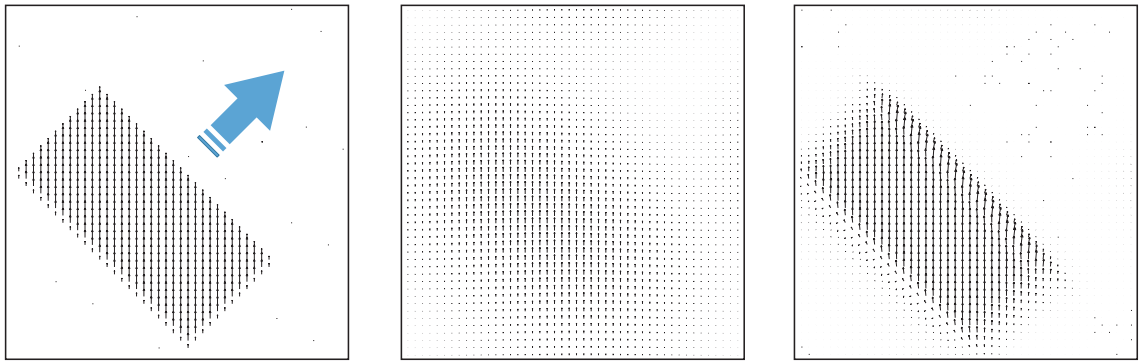
$$\begin{aligned} (\mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega})^x_{i,j} &= -\frac{\Delta t}{h^2} X^y_{i,j} (\mathbf{d}\boldsymbol{\omega})_{i,j-1}, \\ (\mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega})^y_{i,j} &= \frac{\Delta t}{h^2} X^x_{i,j} (\mathbf{d}\boldsymbol{\omega})_{i-1,j}. \end{aligned} \tag{2.13}$$

Note the sign difference is due to the orientation of the extrusions, and would be different if the velocity field changed sign (see figure 2.1). To compute the second half of Cartan's formula we must now compute $\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega}$ at nodes, and then difference them along the edges. Using dimension splitting, as well as averaging the velocity field from edges to get values at nodes, we get for node (i, j),

$$(\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})_{i,j} = \frac{\Delta t}{2h^2} \left((X^x_{i,j} + X^x_{i,j-1}) \boldsymbol{\omega}^x_{i-1,j} + (X^y_{i,j} + X^y_{i-1,j}) \boldsymbol{\omega}^y_{i,j-1} \right). \tag{2.14}$$

We may now trivially compute $\mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega}$ for edges as

$$\begin{aligned} (\mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})^x_{i,j} &= (\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})_{i+1,j} - (\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})_{i,j}, \\ (\mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})^y_{i,j} &= (\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})_{i,j+1} - (\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})_{i,j}. \end{aligned}$$



(a) (b) (c)

Figure 2.4. ***Advection of a 1-Form:*** *(a) A piecewise-constant form (dy within a rectangular shape, 0 outside) is advected in a constant velocity field ($\mathbf{X} = (1, 1)$, blue arrow) on a unit square periodic domain with a grid resolution of 48^2 and a time step $dt = 10^{-3}$. (b) Because the domain is periodic, the form should be advected back to its original position after 1 second (1000 steps); however, our numerical method with a piecewise constant upwind finite volume scheme results in considerable smearing instead. (c) Using a high-resolution scheme (here, WENO-7) as the basic component of our form advection procedure significantly reduces smearing artifacts (same number of steps and step size).*

Cartan's formula and the definition of Lie advection now lead us to obtain

$$\Delta\boldsymbol{\omega} = -\int_0^{\Delta t} \boldsymbol{\mathcal{L}}_{\boldsymbol{X}}\boldsymbol{\omega} dt$$

discretized as the update rule

$$\begin{aligned}\boldsymbol{\omega}^x_{i,j} += -(\mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega})^x_{i,j} - (\mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})^x_{i,j},\\ \boldsymbol{\omega}^y_{i,j} += -(\mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega})^y_{i,j} - (\mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega})^y_{i,j}.\end{aligned}$$

A first example. An example of this low-order scheme can be seen in figure 2.4(a—b) where we advect a piecewise constant 1-form by a constant vector field $\boldsymbol{X} = (1, 1)$ in a periodic domain. Advecting the form forward in this velocity field for a time of 1 second brings the form back to its original position in the continuous case; however, our numerical scheme proves very diffusive, as expected on discontinuous forms. We can however measure the error of our scheme by comparison with initial conditions as a function of the grid resolution with appropriately scaled time step sizes. To measure the error we recall the L_p norm of a k-form $\boldsymbol{\omega}$ is defined over a smooth manifold $\boldsymbol{\mathcal{M}}$ as

$$|\boldsymbol{\omega}|_p = \left[\int_{\boldsymbol{\mathcal{M}}} |\boldsymbol{\omega}|^p d\mu\right]^{1/p}, \quad \text{where} \quad |\boldsymbol{\omega}| = (\boldsymbol{\omega}, \boldsymbol{\omega})^{\frac{1}{2}}_{\boldsymbol{\mathcal{M}}},$$

and $(\cdot, \cdot)_{\boldsymbol{\mathcal{M}}}$ is the scalar product of k-forms defined by the Riemannian metric, and $d\mu$ is its associated volume form. We hence define the 1- and 2-norms of discrete 1-forms on a 2-dimensional regular

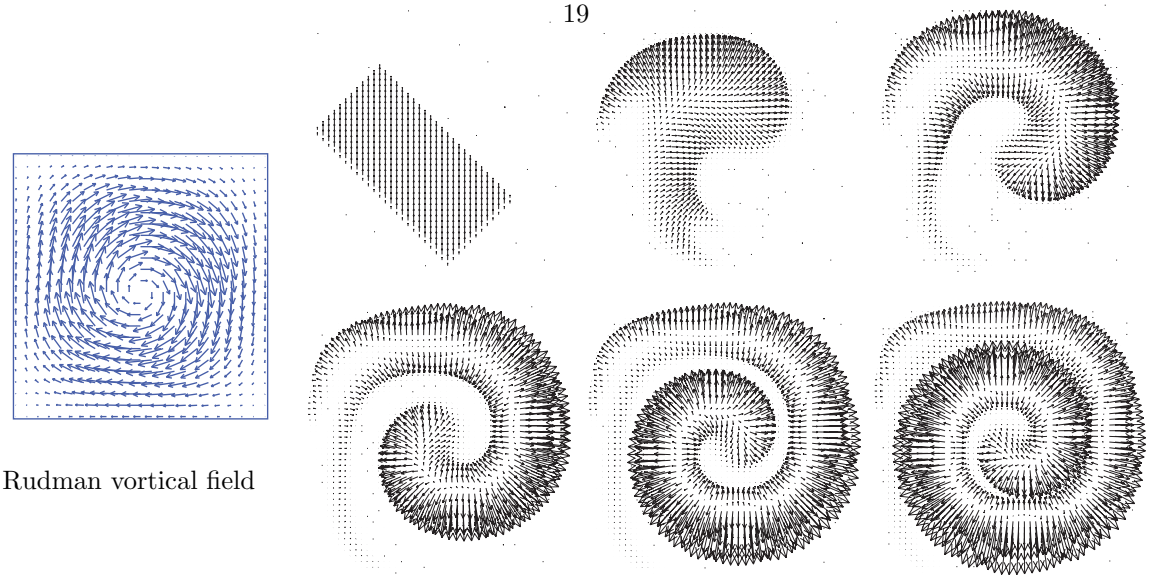


Figure 2.5. ***High-Order Advection:*** *in a vortical vector field (left) typically used for scalar advection, a piecewise-constant form is advected on a unit square periodic domain with a grid resolution of* 48^2 *and a time step* $dt = 10^{-3}$ *for 0, 200, and 400 steps (top), 600, 800, and 1000 steps (bottom).*

grid with spacing h as

$$|\boldsymbol{\omega}|_1 = h \sum_{i,j} (|\boldsymbol{\omega}^x_{i,j}| + |\boldsymbol{\omega}^y_{i,j}|),$$

$$|\boldsymbol{\omega}|_2 = \left(\sum_{i,j} (|\boldsymbol{\omega}^x_{i,j}|^2 + |\boldsymbol{\omega}^y_{i,j}|^2) \right)^{\frac{1}{2}}$$

for simplicity, but we found using more sophisticated discretizations of the norms all yielded similar results. Figure 2.6(c) shows the error plot in L_1 and L_2 norms of this simple example under power-of-two refinement, confirming the first-order accuracy of our approach.

High-resolution methods. Note that had we chosen to leverage more sophisticated finite volume solvers in the previous example, the only changes would occur in equations (2.13) and (2.14), which would use the new numerical flux for computing the discrete contraction: any 2-dimensional method could be used for equation (2.13), while a 1 dimensional method is required for equation (2.14). Due to the dimensional splitting obtaining higher-order schemes is not easy, but for many application the order of accuracy is not always the most important thing. In particular, in the presence of discontinuous solutions *high-resolution* methods are often preferred for their ability to better preserve discontinuities and reduce diffusion. To test the utility and effectiveness of such schemes applied to forms, we compare the piecewise-constant upwinding method from the previous section with a Finite Volume 7^{th}-order 1-dimensional WENO upwind scheme (see an overview of FV-WENO methods in [46]). Figure 2.4(c) shows the high-resolution finite volume scheme does a much better

job at preserving the discontinuities, despite both methods being of the same order of accuracy for this discontinuous initial form (figure 2.6(c)).

Accuracy. To further demonstrate that properties from the underlying finite volume schemes chosen (including their accuracy) carry over to the advection of forms, we provide additional numerical tests. In figure 2.5, we advected a simple discontinuous 1-form in a vortical shearing vector field (Rudman vortex, left) on a 48x48 grid representing a periodic domain. As expected, the form is advected in a spiral-like fashion. By advecting the shape back in the the negated velocity field for the same amount of time, we can derive error plots to compare the L_1 and L_2 norms for this example under refinement of the grid; see figure 2.6 for other convergence tests when a first-order upwind method or a WENO-7 method is used in our numerical technique.

2.4.3 Properties

It is easy to show that our discrete Lie derivative will commute with the discrete exterior derivative as in the continuous case, by using Cartan's formula and a discrete exterior derivative that satisfies $\mathbf{dd} = 0$ since we have

$$\begin{aligned}
\mathbf{d}\boldsymbol{\mathcal{L}}_{\boldsymbol{X}}\boldsymbol{\omega} &= \mathbf{d}(\mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega} + \mathbf{d}\mathbf{i}_{\boldsymbol{X}}\boldsymbol{\omega}) \\
&= \mathbf{d}\mathbf{i}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega} \\
&= (\mathbf{d}\mathbf{i}_{\boldsymbol{X}} + \mathbf{i}_{\boldsymbol{X}}\mathbf{d})\mathbf{d}\boldsymbol{\omega} \\
&= \boldsymbol{\mathcal{L}}_{\boldsymbol{X}}\mathbf{d}\boldsymbol{\omega}.
\end{aligned}$$

This commutativity does not depend on any properties of the discrete contraction and therefore holds regardless of the underlying finite volume scheme chosen. A useful consequence of this fact is that *the discrete Lie advection of closed forms will remain closed by construction*; i.e., the advection of a gradient (resp., curl) field will remain a gradient (resp., curl) field.

Unfortunately other properties of the Lie derivative do not carry over to the discrete picture as easily. The product rule for wedge products, for example, does not hold for the discrete wedge products defined in [13], although perhaps a different discretization of the wedge product may prove otherwise. However, the nonlinearity of the discrete contraction operator along with the upwinding potentially picking different directions on simplices and their subsimplices makes designing a discrete analog satisfying this continuous property challenging.

2.5 Conclusions

In this chapter we have introduced an extension of classical finite volume techniques for hyperbolic conservation laws to handle arbitrary discrete forms. A class of first-order finite-volume-based

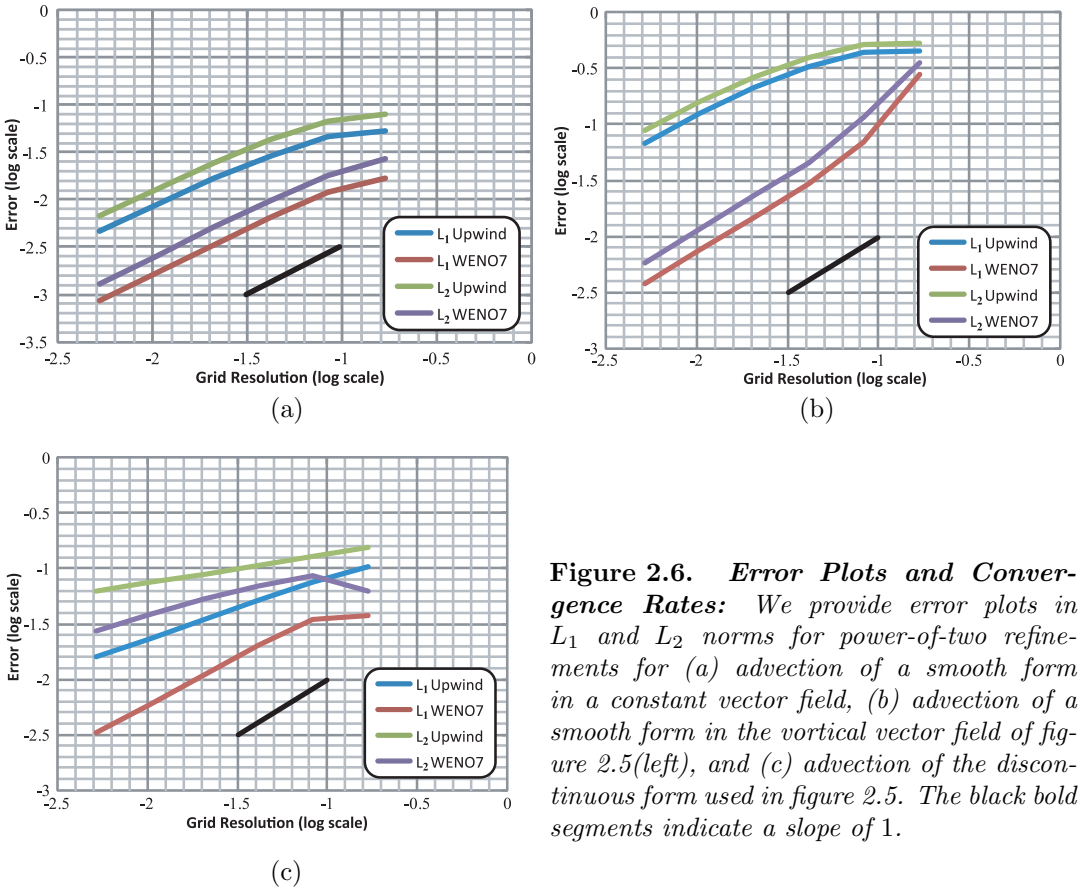


Figure 2.6. ***Error Plots and Convergence Rates:*** *We provide error plots in L_1 and L_2 norms for power-of-two refinements for (a) advection of a smooth form in a constant vector field, (b) advection of a smooth form in the vortical vector field of figure 2.5(left), and (c) advection of the discontinuous form used in figure 2.5. The black bold segments indicate a slope of* 1.

discretizations of both contraction and Lie derivative of arbitrary forms was presented, extending Discrete Exterior Calculus to include approximations to these operators. Low numerical diffusion is attainable through the use of high-resolution finite volume methods. The advection of forms and vector fields are applicable in a multitude of problems, including conservative interface advection and conservative vorticity evolution.

Although finite volume methods can offer high resolution at a relatively low computational cost, numerical diffusion is still present and can accumulate over time. In addition, the numerical scheme we presented is not variational in nature, i.e., it is not (a priori) derived from a variational principle. These limitations are good motivations for future work.

While we have given numerical evidence demonstrating the manner in which resolution and accuracy is inherited from the underlying finite volume scheme, a formal analysis of stability and convergence remains to be performed. In particular, it is desirable to understand what the stability of the underlying one-dimensional scheme implies about the stability of our method. Although norms for discrete differential forms have been defined, such tools are not always suitable for the analysis of nonlinear methods in multiple dimensions, even for scalar advection. We believe such an investigation is an important next step for the present work.

In the future, we also expect that extensions can be made to make truly high-order and high-

resolution discretizations of the contraction and Lie derivative through n-dimensional reconstructions of k-forms and extrusions. In particular, this would greatly facilitate the extension to simplicial meshes. While there has been recent progress on high-order schemes for triangular meshes [47, 48], these are not directly applicable for contractions of arbitrary forms. Hence, despite having a well-defined discrete exterior derivative for simplicial meshes, such an extension will require more than just 0- or n-form advection schemes, as our current dimension splitting approach to generalize such schemes for contraction of forms of other degrees does not immediately apply on nonrectangular meshes.

Chapter 3

Hodge-Optimized Triangulations

3.1 Introduction

The discrete Lie advection introduced in the previous chapter was developed for regular grids, as its extension to simplicial meshes is not as easy. However, the traditional DEC operators can be used directly on simplicial meshes as well as regular grids, and nontrivial boundaries often call for such spatial discretizations. A vast array of modeling and simulation techniques assume that a mesh is given, providing a discretization of a 2-dimensional or 3-dimensional domain in simple triangular or tetrahedral elements. As the accuracy and stability of most computational endeavors heavily depend on the shape and size of the worst element [49], mesh element quality is often a priority when conceiving a mesh generation algorithm. Be it for finite-volume, finite-element, finite-difference, or less mainstream computational schemes, the need for good triangle or tetrahedron meshes is ubiquitous not only in computer graphics, but in computational sciences as well—and as computational power increases, so does the demand for effective meshing.

While generically "good" dual or primal elements can be obtained via Centroidal Voronoi Tesselations [50] or Optimal Delaunay Triangulation [51] respectively, an increasing number of numerical methods need strict control over *both* primal and dual meshes: from discrete differential operators in modeling (e.g., [52]) to pressure solves in fluid simulation (as recently mentioned in [53]), the placement of primal elements with respect to their orthogonal dual elements is increasingly recognized as crucial to reliable computations. However, very little is available to quickly and effectively design such orthogonal primal-dual structures over complex domains. To address this lack of adequate meshing tools, we introduce a theoretical analysis of what makes a mesh and its dual numerically optimal in some common graphics contexts, along with practical algorithms to produce optimized primal-dual triangulations.

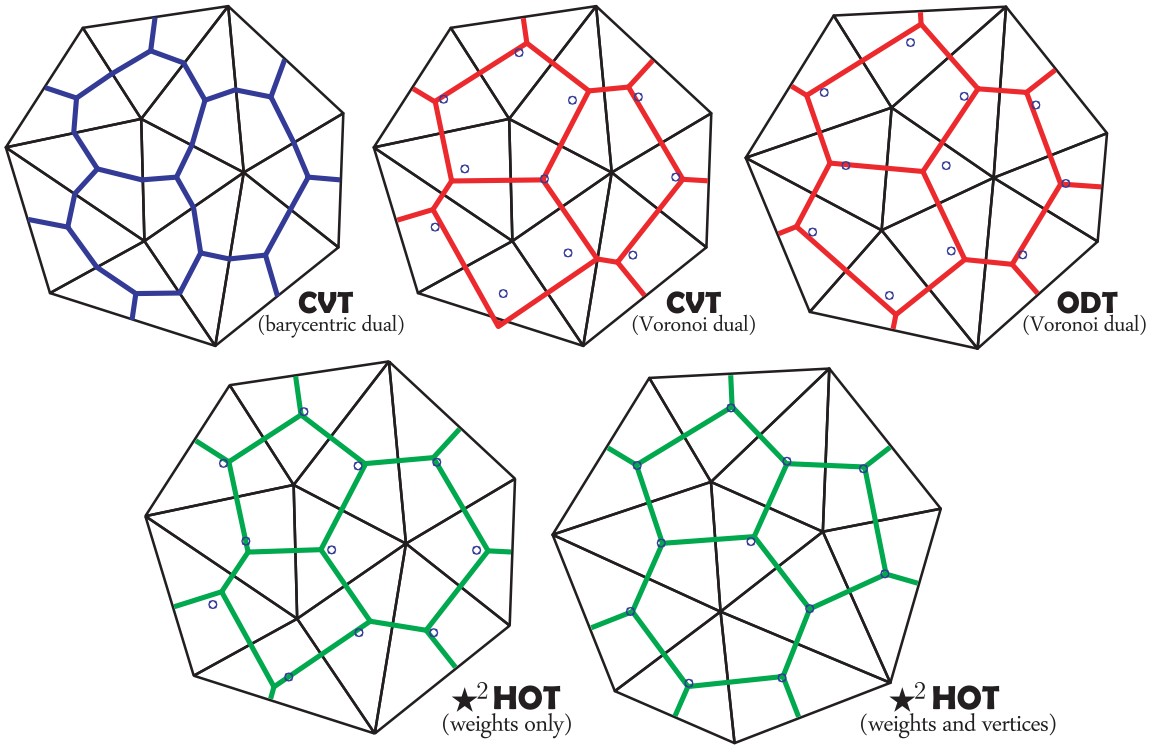


Figure 3.1. ***Primal/Dual Triangulations****: Using the barycentric dual (top-left) does not generally give dual meshes orthogonal to the primal mesh. Circumcentric duals, both in Centroidal Voronoi Tesselations (CVT, top-middle) and Optimal Delaunay Triangulations (ODT, top-right), can lead to dual points far from the barycenters of the triangles (blue points). Leveraging the freedom provided by weighted circumcenters, our Hodge-optimized triangulations (HOT) can optimize the dual mesh alone (bottom-left) or both the primal and dual meshes (bottom-right), e.g., to make them more self-centered while maintaining primal/dual orthogonality.*

3.1.1 Previous Work

Meshing complex 2-dimensional or 3-dimensional domains with high-quality elements has generated a tremendous number of research efforts. Bounds on numerical errors have resulted in the use of Delaunay triangulations [54] for finite-element computations, and Voronoi diagrams [55] for finite-volume methods. However, the combined use of a primal mesh and its dual structure has increased over the last decade in both modeling and simulation, with quantities of both geometric (normals, mean and Gaussian curvatures, tangents) and physical (velocities, fluxes, circulations, vorticities) nature inherently located either on the primal mesh or its dual [56]. Calculations involving these primal and dual values in graphics were formalized in Discrete Exterior Calculus (DEC—see section 1.2), now used in vision and image processing as well [57].

Delaunay/Voronoi pairs. In the context of discrete differential geometric operators, Meyer et al. [52] recommended a Voronoi (circumcentric) dual for tighter error bounds—but locally reverted to the barycentric dual when a dual vertex was not contained in its primal simplex. For fluid

simulation, Perot and Subramanian [58] and Elcott *et al.* [28] advocated circumcentric duals as well, this time to ensure that pressure gradients between adjacent cells were parallel to the velocity samples stored on the common face. In DEC terminology, this simply means that the flux through a face and the circulation along its associated dual edge measure the same component of a vector field. Moreover, another advantage of the Delaunay/Voronoi duality for fluid simulation exploited in [28] is that the convexity and non-self-intersection of dual Voronoi cells make them ideal for the use of generalized barycentric coordinates [59]. Still, the seemingly natural choice of Delaunay/Voronoi triangulation is far from being without drawbacks. First and foremost, it it extremely difficult in practice to get "self-centered" Delaunay triangulations [60] for which each circumcenter lies inside its associated tetrahedron: failure to satisfy this property locally can lead to numerical degeneracies. Recent methods attempting to optimize meshes to avoid this issue remain impractical for complex domains [61]. A second drawback of a Delaunay/Voronoi pair is the inability to choose the positions of dual nodes locally without significantly degrading the primal mesh: having more flexibility in the placement of pressure samples would significantly improve the treatment of free surfaces in embedded boundary methods [53]. Consequently, and while abundantly vetted by theoretical guarantees, Delaunay/Voronoi triangulations are too restrictive in many practical situations. We will, instead, promote the use of arbitrary convex orthogonal primal/dual pairs to offer significantly more flexibility (see figure 3.1).

Accuracy vs efficiency. Sparsity is crucial when dealing with large linear algebraic problems frequently encountered in geometry processing. Graphics literature is replete with low-order methods using as-sparse-as-possible formulations for efficiency. While nonlinear and/or high-order methods have their own advantages and proponents, it is often highly desirable to find the simplest, fastest approximation valid for most applications, even if only to initialize a subsequent nonlinear solver. In the context of DEC, this quest for efficiency often translates to the use of the so-called diagonal Hodge stars (which include the famous cotangent weights [62] widely used in geometry processing) over other discretizations of Hodge stars [63, 64, 65] to approximate primal values based on dual values (and vice versa); as inverse Hodge stars appear even in basic operators [66, 28], having diagonal approximations can greatly increase computational efficiency. However, once a primal-dual triangulation is chosen, one has no control over the error incurred by diagonal approximations: efficiency may thus only be achieved on particularly good meshes. We will, instead, design meshes to minimize formal error bounds of diagonal Hodge stars, generally increasing accuracy without the additional costs associated with refinement (increasing system sizes) or higher-order Galerkin Hodge stars (decreasing the sparsity and making inversion more difficult).

3.2 Preliminaries and Definitions

Before introducing our Hodge-based meshes, we first provide some background on orthogonal primal/dual triangulations and optimal transport as we will make heavy use of these notions throughout the chapter.

3.2.1 Regular-Power Triangulations

The notion of dual for a triangulation $\mathcal{T}$ in $\mathbb{R}^d$ is well-known and routinely used in graphics: each d-simplex is associated with a dual vertex (dual 0-cell), each $(d-1)$-simplex is associated with a dual edge (dual 1-cell) between the two dual vertices associated with the two adjacent d-simplices, etc. Primal vertices $\mathbf{x}_i$ are then associated with dual d-cells V_i, and the dual of $\mathcal{T}$ forms a cell complex $\mathcal{D}$. However, this concept of dual is abstract, as the location of the dual vertices are not specified a priori. A very common dual to a triangulation is the cell complex that uses the circumcenters of each d-simplex as dual vertices. If the initial triangulation is Delaunay (i.e., satisfying the empty circumsphere property [54]), this dual is simply the Voronoi diagram of the primal vertices, and its nice properties of non-self-intersection, convexity, and orthogonality of the primal and dual elements have led to its use in countless papers in graphics and computational sciences. The barycentric dual, for which barycenters are used instead of circumcenters (see figure 3.1), is also quite common in particular for finite-volume computations; however, it fails to satisfy both the orthogonality and the convexity conditions on general triangulations.

Desirable primal-dual pairs. Remaining agnostic with respect to the choice of a dual, we will call a **primal-dual triangulation** $\mathcal{M}$ in $\mathbb{R}^n$ any pair $(\mathcal{T}, \mathcal{D})$ with $\mathcal{T}$ being a triangulation in $\mathbb{R}^d$ and $\mathcal{D}$ a compatible dual complex of $\mathcal{T}$ (i.e., their respective adjacency matrices are transpose of each other). Moreover, if every edge $[\mathbf{x}_i, \mathbf{x}_j]$ of $\mathcal{T}$ and its *dual* $V_i \cap V_j$ in $\mathcal{D}$ are orthogonal to each other, the pair $(\mathcal{T}, \mathcal{D})$ is said to form an *orthogonal primal-dual triangulation*. Finally, we will denote as $*$ the operation of **duality** (see figure 3.2); that is, a primal simplex σ will have its dual referred

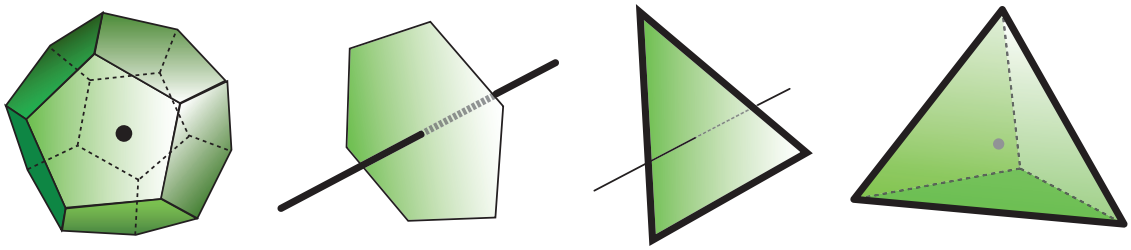


Figure 3.2. ***Duality****: The dual of a triangulation in $\mathbb{R}^d$ associates to each k-simplex σ^k a $(d-k)$-cell $*\sigma^k$ (here, $k = 0, 1, 2, 3$). Having $\sigma^k \cap *\sigma^k$ "well centered" within the primal simplex* and *its orthogonal dual cell is crucial to numerics in modeling and simulation.*

to as $*\sigma$ with the orientation induced by the primal orientation and the manifold orientation. For a more formal definition, see [67, 13].

Regular/power duality. Delaunay/Voronoi primal-dual triangulations are restrictive in that they allow no choice on the dual once the primal mesh is fixed. A natural question to ask is are there other primal-dual triangulations that satisfy the orthogonality, non-intersection, and convexity conditions we require? The answer is affirmative: the known duality between **regular triangulations** (also called weighted Delaunay triangulations) and **power diagrams** (also called Laguerre or weighted Voronoi diagrams) provides *all* the satisfactory orthogonal primal-dual triangulations [68]. This exact characterization of the primal/dual triangulations we seek will be particularly convenient as it will lead to a number of new geometric functionals measuring mesh quality; it will also yield straightforward generalizations of standard DEC operators without some of the most limiting factors that the Delaunay/Voronoi duality possesses.

Formally, a weighted point set is defined as a pair $(\mathbf{X}, W)=\{(\mathbf{x}_1, w_1), \ldots, (\mathbf{x}_n, w_n)\}$, where $\mathbf{X}$ is a set of points in $\mathbb{R}^d$, and $\{w_i\}_{i\in[1,\ldots,n]}$ are real numbers called weights. The **power** of a point $\mathbf{x} \in \mathbb{R}^d$ with respect to a weighted point $(\mathbf{x}_i, w_i)$ (sometimes referred to as the Laguerre distance) is defined as $\|\mathbf{x}-\mathbf{x}_i\|^2-w_i$, where $\|.\|$ stands for the Euclidean distance. Using this power definition, to each $\mathbf{x}_i$ we can associate its weighted Voronoi region $V_i^w=\{\mathbf{x}\in\mathbb{R}^d|\ \|\mathbf{x}-\mathbf{x}_i\|^2-w_i \leq \|\mathbf{x}-\mathbf{x}_j\|^2-w_j, \forall j\}$. The power diagram of $(\mathbf{X}, W)$ is the cell complex whose cells are the weighted Voronoi regions. The dual of the power diagram of $(\mathbf{X}, W)$ is the regular triangulation of $(\mathbf{X}, W)$: this triangulation contains a k-simplex with vertices $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k$ in $\mathbf{X}$ iff $\bigcap_{j=0}^{j=k} V_j^w \neq \emptyset$.

Note that in a regular triangulation, a point $\mathbf{x}_i \in \mathbf{X}$ can be hidden, i.e., it may not be used in the triangulation because its weighted Voronoi region is empty. Note also that when the weights are all equal, the power diagram coincides with the Euclidean Voronoi diagram of $\mathbf{X}$. Geometrically, one can think of the weight w_i as the square of the radius of a unique circle centered at vertex $\mathbf{x}_i$; then there exists in each triangle a circle, centered at what is known as the **weighted circumcenter**, which is orthogonal to each of the circles centered at the vertices. All of these properties can be extended to the case where the weights are negative [69], and thus regular triangulations and their associated power diagrams generalize the usual Delaunay/Voronoi duality nicely. This simple addition of a weight to each vertex allows us to conveniently parameterize the *entire* space of good (i.e., orthogonal, convex, and non-self-intersecting) primal-dual triangulations $\mathcal{M}$.

3.2.2 Basics of Optimal Transport

The optimal transport problem dates back to Gaspard Monge. (For a description of the vast literature on this topic, we refer the reader to [70].) In essence, it seeks to determine the optimal way to move a pile of dirt M to a hole N of the same volume, where "optimal" means that the integral

of the distances by which the dirt is moved (one infinitesimal unit of volume at a time) is minimal. While Monge's variational formulation of the problem assumed that all the dirt at a point $x \in M$ must be moved through a point-to-point mapping s to one location $s(x) \in N$, this restriction was relaxed by Kantorovich who replaced the mapping s with a probability measure $\pi \in \mathcal{P}(M \times N)$ that specifies the joint measure of dirt-hole correspondences; i.e., π is a *transport plan* between a probability measure μ on M and a probability measure ν on N with $\pi(\cdot \times N) = \mu$ and $\pi(M \times \cdot) = \nu$. This extension to the transport of measures marked a renewed interest in optimal transport as it proved general enough to apply to many scientific fields (for recent graphics-related applications, see [71, 72]).

Wasserstein metric. For measures the notion of "distance" (i.e., cost of transport) may vary based on context. A common distance function defined between probability measures in $\mathbb{R}^d$ with bounded support is the q-Wasserstein metric, defined as

$$W_q(\mu, \nu) = \left(\inf_{\pi \in \mathcal{P}(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \| x - y \|^q \, \mathrm{d}\pi(x, y) \right)^{1/q}.$$

To reuse the analogy mentioned above, if each measure is viewed as a unit amount of piled-up dirt, the metric is the minimum "cost" of turning one pile into the other, which is assumed to be the amount of dirt that needs to be moved times the L_p distance it has to be moved. Because of this analogy, the metric is sometimes called the earth mover's distance. Note, as it will be crucial in section 3.3, that by a direct application of the Hölder inequality for two measures of unit mass,

$$W_1(\mu, \nu) \leq W_2(\mu, \nu). \tag{3.1}$$

Finally, we will also need the Kantorovich-Rubinstein theorem, stating that for two measures μ and ν with bounded support, the 1-Wassertein distance between μ and ν can be rewritten as

$$W_1(\mu, \nu) = \sup_{\substack{\varphi : \mathbb{R}^d \to \mathbb{R} \\ \mathrm{Lip}(\varphi) \leq \lambda}} \frac{1}{\lambda} \int_{\mathbb{R}^d} \varphi(x) \, \mathrm{d}(\mu - \nu), \tag{3.2}$$

where $\mathrm{Lip}(\varphi)$ represents the Lipschitz constant of function φ. This expression will be useful shortly to link optimal transport and approximation error of diagonal Hodge stars.

3.3 Error Functionals for Diagonal Hodge Stars

To demonstrate the advantages of using regular/power triangulations, we focus on a particularly relevant type of functional measuring primal and dual properties. Recall from section 1.2 that for an arbitrary primal element σ, the diagonal approximation of the Hodge star $\star$ [63] of a continuous

differential form α assumes

$$\int_{*\sigma} \star\alpha \approx \frac{|*\sigma|}{|\sigma|} \int_{\sigma} \alpha, \tag{3.3}$$

where $|.|$ denotes the Lebesgue measure (length, area, volume) of a simplex or cell. In other words, the discrete kth Hodge star is encoded as a diagonal matrix $\star^k$ with

$$\forall i, \quad (\star^k)_{ii} := \frac{|*\sigma_i^k|}{|\sigma_i^k|},$$

where σ_i^k (resp., $*\sigma_i^k$) is the ith k-simplex (resp., $(d-k)$-cell) of the primal-dual triangulation $\mathcal{M} = (\mathcal{T}, \mathcal{D})$; the discrete Hodge star of a discrete primal k-form ω^k is then computed as $\star^k \omega^k$, and the extension to dual discrete forms (now with $(\star^k)^{-1}$) is trivial (for further details see, e.g., [56]).

3.3.1 Deriving Tight Bounds through Optimal Transport

While computationally convenient, diagonal Hodge stars are not very accurate: they are generally only exact for constant forms. We can quantify the induced inaccuracy of $\star^k$ by defining the *error density* e_i on the dual of a k-simplex σ_i as the average difference between the discrete approximation and the exact Hodge star value:

$$e_i := \frac{1}{|*\sigma_i|} \left| \frac{|*\sigma_i|}{|\sigma_i|} \int_{\sigma_i} \omega - \int_{*\sigma_i} \star\omega \right| = \left| \frac{1}{|\sigma_i|} \int_{\sigma_i} \omega - \frac{1}{|*\sigma_i|} \int_{*\sigma_i} \star\omega \right|.$$

We now notice that due to the orthogonality of σ and $*\sigma$, the component of ω along σ is the same as the component of $\star\omega$ along $*\sigma$ (this is the same property that allows orthogonal primal-dual triangulations to admit a diagonal Hodge star in the first place). Writing this component as a scalar function $f(x)$, we can rewrite the integrals involved above as

$$\int_{\sigma_i} \omega \; = \int_{\sigma_i} f(x) \; d\mu_{\sigma_i} \quad \text{and} \quad \int_{*\sigma_i} \star\omega \; = \int_{*\sigma_i} f(x) \; d\mu_{*\sigma_i},$$

where $d\mu_{\sigma_i}$ and $d\mu_{*\sigma_i}$ are the volume forms of σ_i and $*\sigma_i$ respectively. We can use these expressions to rewrite the error density as

$$e_i = \left| \int_{\sigma_i} f(x) \frac{d\mu_{\sigma_i}}{|\sigma_i|} - \int_{*\sigma_i} f(x) \frac{d\mu_{*\sigma_i}}{|*\sigma_i|} \right| = \left| \int_{\mathbb{R}^d} f(x) \, d(\bar{\mu}_{\sigma_i} - \bar{\mu}_{*\sigma_i}) \right|, \tag{3.4}$$

where now $d\mu_{\sigma_i}/|\sigma_i|$ and $d\mu_{*\sigma_i}/|*\sigma_i|$ are uniform probability distributions over σ_i and $*\sigma_i$ respectively, and $d\bar{\mu}_{\sigma_i}$ and $d\bar{\mu}_{*\sigma_i}$ are their trivial extensions to $\mathbb{R}^d$—i.e., for any measurable set S in $\mathbb{R}^d$,

$$\int_S d\bar{\mu}_{\sigma_i} = \int_{S \cap \sigma_i} \frac{d\mu_{\sigma_i}}{|\sigma_i|} \quad \text{and} \quad \int_S d\bar{\mu}_{*\sigma_i} = \int_{S \cap *\sigma} \frac{d\mu_{*\sigma_i}}{|*\sigma_i|}.$$

From equation (3.4) and equation (3.2), we deduce that the *tightest* bound one can find on the Hodge star error density per simplex for an arbitrary λ-Lipschitz form is simply λ times the minimum cost over all transport plans between σ_i (seen as a uniform probability measure over the mesh element) and $*\sigma_i$ (also seen as a uniform probability measure over the dual element); that is, with a slight abuse of notation,

$$e_i \leq \lambda \, W_1(\sigma_i, *\sigma_i). \tag{3.5}$$

This formally establishes a link between Hodge star accuracy and optimal transport. Note that we only required ω to be Lipschitz continuous, a reasonable assumption in many graphics applications.

3.3.2 Error Functionals on Meshes

From these local error densities, we can assemble a total error by taking the $L_{p\geq 1}$ integral norm of the error over the mesh area, i.e., by summing the integrals of the pth power of the error densities e_i over local regions, specific to σ_i and $*\sigma_i$, that tile the mesh. Such regions have been defined in previous work, coined "support volumes" in [13] and "diamonds" in [73, 56]: when σ_i and $*\sigma_i$ intersect, these regions, which we will refer to as $\diamond(\sigma_i \cup *\sigma_i)$, are just the convex hulls of σ_i and $*\sigma_i$; in the general case, they become signed unions of convex hulls of the primal vertices of σ_i and each boundary element of the dual cell $*\sigma_i$. Using Σ^k to denote the set of k-simplices of a triangulation, the total error is thus

$$E_p(\mathcal{M}, \star^k) = \left(\sum_{\sigma_i \in \Sigma^k} \int_{\diamond(\sigma_i \cup *\sigma_i)} {e_i}^p \right)^{\frac{1}{p}} = \left(\sum_{\sigma_i \in \Sigma^k} \frac{|\sigma_i||*\sigma_i|}{\binom{d}{k}} \, {e_i}^p \right)^{\frac{1}{p}},$$

since the volume of the diamond $\diamond(\sigma_i \cup *\sigma_i)$ is, up to a dimension factor, simply the product of the primal and dual volumes due to our primal/dual orthogonality assumption of mesh $\mathcal{M}$.

From equation (3.5), we conclude that a tight bound for the pth power of the total error is expressed as

$$E_p(\mathcal{M}, \star^k)^p \leq \frac{\lambda^p}{\binom{d}{k}} \sum_{\sigma_i \in \Sigma^k} |*\sigma_i||\sigma_i| \, W_1(\sigma_i, *\sigma_i)^p. \tag{3.6}$$

Notice that $E_\infty(\mathcal{M}, \star^k)$ is thus, up to the Lipschitz constant, bounded by the maximum of the W_1 distances between primal and dual elements of the mesh as expected. For notational convenience, we will denote by $\star^k\text{-HOT}_{p,1}(\mathcal{M})$ the bound (with Lipschitz and dimension constants removed) obtained in equation (3.6); more generally, we will define

$$\boxed{\star^k\text{-HOT}_{p,q}(\mathcal{M}) \equiv \sum_{\sigma_i \in \Sigma^k} |*\sigma_i||\sigma_i| \, W_q(\sigma_i, *\sigma_i)^p}$$

as relevant functionals (or energies) to construct meshes, since minimizing them will control the

quality of the discrete Hodge stars.

Continuity of HOT functionals. Because they are based on volume integrals, the HOT functionals are continuous over the space of regular/power triangulations. They are indeed continuous in the vertex positions of the primal and dual meshes, but also through primal mesh flips: an edge or face flip in a regular triangulation happens when a dual (power) edge vanishes. Hence the diamond weighting we use for our total error renders our HOT functionals continuous with respect to both vertices and weights. This will be particularly convenient when it comes time to optimize a mesh in order to minimize these functionals.

3.3.3 Discussion

Our HOT energies are archetypical, general-purpose examples of mesh quality measures imposed on both primal *and* dual meshes, but they are by no means unique: from the local error densities e_i, other energies can be formulated to target more specific errors occurring in mesh computations (see some examples in section 3.5). In particular, linear combinations of HOT energies may be used if multiple Hodge stars are needed, for example when using Laplacians of k-forms with $k > 1$. Note also that the use of a 1-Wasserstein distance is notably less attractive numerically than a 2-Wasserstein distance as we will discuss in section 3.4.4. Fortunately, we can also provide a bound of the Hodge star error, which, while less tight than the previously derived $\mathrm{HOT}_{p,1}$, will be particularly convenient to deal with computationally: the existence of optimal transport plans when the cost is the distance squared (i.e., W_2) being well studied, a useful bound on the Hodge star error can be derived using the inequality given in equation (3.1) to yield

$$E_2(\mathcal{M}, \star^k)^2 \leq \sum_{\sigma_i \in \Sigma^k} |*\sigma_i||\sigma_i| \; W_2(\sigma_i, *\sigma_i)^2 \equiv \star^k\text{-}\,\mathrm{HOT}_{2,2}(\mathcal{M}).$$

The reader may have noticed that the functional $\star^0\text{-}\,\mathrm{HOT}_{2,2}(\mathcal{M})$ is, in the case of equal weights, the well-known Centroidal Voronoi Tesselation (CVT) energy ($\sum_i \int_{V_i} \|\mathbf{x}-\mathbf{x}_i\|^2 dV$) for which several minimization techniques, from Lloyd iterations [50] to quasi-Newton methods [74], have been developed. L_p variants (i.e., $\star^0\text{-}\,\mathrm{HOT}_{2p,2}(\mathcal{M})$ for $p \geq 2$) were also explored recently [75]. However, these energies only correspond to $\star^0$, and are not as tight as $\mathrm{HOT}_{1,p}$. Our HOT energies can thus be seen as a direct generalization of the CVT-like functionals. Note finally that the Optimal Delaunay Triangulation (ODT) energy used in [51] can also be seen as a variant of $\star^d\text{-}\,\mathrm{HOT}_{2,2}(\mathcal{M})$ in $\mathbb{R}^d$ for which the dual mesh is restricted to be "barycentric"; alas, the resulting mesh will not necessarily lead to an orthogonal primal-dual triangulation—even if the resulting simplices were proven to be very close to isotropic.

3.4 Hodge-Optimized Triangulations

In the remainder of this chapter, we call a HOT triangulation any pair $\mathcal{M}$ consisting of a regular triangulation $\mathcal{T}$ and its associated power diagram $\mathcal{D}$ for which $\mathcal{T}$, $\mathcal{D}$, or both, have been optimized in order to reduce one (or a linear combination of) HOT functional(s). We now describe the basic computations involved in optimizing meshes for two particularly interesting (and unexplored) families of energies: $\mathrm{HOT}_{2,2}$ and $\mathrm{HOT}_{1,1}$.

3.4.1 General Minimization Procedure

Given that both (continuous) vertex positions and (discrete) mesh connectivity need to be optimized, the task of finding HOT meshes is seemingly intractable. Thankfully, regular triangulations provide a good parameterization of the type of primal-dual meshes we wish to explore: one can simply optimize the continuous values of both positions and associated weights to find a HOT mesh. However, HOT energies are not convex in general, and a common downfall of nonconvex optimization is its propensity to settle into local minima. In our case, finding a good nonoptimal minimum is often enough to dramatically improve the mesh quality. We thus start our minimization process by initializing the domain with uniformly sampled vertices over the domain, and running a few iterations of CVT [50] or ODT [51] to quickly disperse the vertices and get mesh elements roughly similar in size: from such a decent initial mesh, an optimized mesh can be quickly obtained by performing a gradient descent, or alternatively (without much added implementation complexity), an L-BFGS algorithm [76]—a particular quasi-Newton method where the (inverse) Hessian is approximated based on the M previous steps (we use $M = 7$). A (binary or golden-ratio) linear search is performed to adapt the step size along the gradient or the quasi-Newton direction based on two simple tests (known as Wolfe conditions): the step size should be small enough to make sure the energy decreases, but large enough to induce a marked gradient change. This common minimization procedure works quite well without requiring anything else but an evaluation of our HOT energies and their gradients, which we will derive in closed form from direct integration and/or application of the Reynolds theorem. Note finally that the positions $\mathbf{x}_i$ and the weights w_i have very different scales (units of m vs m^2); we thus proceed by alternatively minimizing our HOT energies with respect to vertex positions and weights. After each step the connectivity is updated using the 2-dimensional or 3-dimensional regular triangulation package of CGAL [77]. Pseudocode of our general procedure is given in figure 3.3, but more specialized optimization techniques could most likely be devised; in particular, based on the HOT energy we wish to minimize, a few alternative minimization procedures may be simpler to implement or faster to converge. We will point out some such special cases shortly.

While both position *and* weight are optimized by default, HOT optimizations are relevant even if only one of these optimizations is performed. For instance, if one has a given (possibly nonflat)

triangulation, vertices could be held fixed while weights are optimized to better one or more of the Hodge stars. Similarly, weights could be kept fixed, e.g., in contexts where they represent power or capacity of the nodes, and a best node placement is sought after—or simply in cases where a given connectivity needs to be maintained. We will discuss some useful variants in section 3.5.

Boundary handling. As in any variational method, boundary conditions can significantly affect the results. Except for the work of Alliez et al. [51, 78, 79], we found very little about boundary handling in previous related work in graphics; for instance, recent papers focusing on the CVT energy like [75, 74] only discuss how to partition a given domain into well-shaped Voronoi cells, providing no insight on dealing with the difficult issue of generating good simplices at the domain boundary. While boundary treatment may be context dependent (fixing vertices or even weights [80] at the boundary being two of the most desirable options), we experimented with a very simple procedure to handle boundaries gracefully for all Hodge stars. We first make sure that each dual vertex $\mathbf{c}$ of a boundary d-simplex T is associated with a "ghost" dual vertex $\hat{\mathbf{c}}$ used to enforce that dual edges at the boundary never have negative lengths: $\hat{\mathbf{c}}$ is put at the projection of $\mathbf{c}$ onto the boundary face of T if $\mathbf{c}$ is within T, and put on top of $\mathbf{c}$ otherwise. We also alter the definition of the energy to become $\mathrm{HOT}\,/|\mathcal{M}|$, i.e., we simply divide the energy by the total area: as volume-shrinkage is no longer rewarded, minimizing the HOT "volume density" makes the optimization steps behave well even at the boundary. We left the evaluations performed to check the Wolfe conditions unchanged: we did not alter the connectivity of the current triangulation (for efficiency reasons) or clamp vertices to the domain during the line search. Once the Newton step is done, however, a vertex is reprojected on the domain's boundary if (a) the vertex has an unbounded weighted Voronoi region, (b) it is outside the domain, or (c) it has one or more of its adjacent triangles' circumcenters *or* barycenters outside the domain. This approach is simple and it lets the vertices move freely from the inside to the boundary and vice versa. We will not incorporate the term $|\mathcal{M}|$ in our derivations (to avoid cluttering the explanations) as it is a trivial alteration of our procedure.

3.4.2 Weighted Circumcenters

Most of the derivations involved in this section rely on the crucial relation linking primal and dual vertices in regular triangulations. We will use $\mathbf{c}(\sigma)$ to denote the weighted circumcenter of simplex σ, i.e., the unique intersection of the mutually-orthogonal affine spaces supporting the primal simplex σ^k and its weighted dual $*\sigma^k$ (see figure 3.4). Of particular importance are the circumcenters of the d-simplices for a mesh $\mathcal{T}$ in $\mathbb{R}^d$: these form the vertices of its (weighted) dual complex $\mathcal{D}$. For a

// MESH OPTIMIZATION
// *Input: vertices* $\mathbf{x}^0 = \{\mathbf{x}_i\}$ *and weights* $w^0 = \{w_i\}$,
// *and a HOT functional* $E(\mathbf{x}, w)$.
$n \leftarrow 0$
repeat
  Compute $E(\mathbf{x}^n, w^n)$ // *See Appendices B.1 and B.2*
  // *Optimize* $\mathbf{x}$
  Pick step direction $\mathbf{d}^x$ for $E(\mathbf{x}^n, w^n)$
  Find α satisfying Wolfe's condition(s)
  $\mathbf{x}^{n+1} \leftarrow \mathbf{x}^n + \alpha\, \mathbf{d}^x$ // *Vertex updates*
  Update regular triangulation
  // *Optimize* w
  Pick step direction $\mathbf{d}^w$ for $E(\mathbf{x}^{n+1}, w^n)$
  Find β satisfying Wolfe's condition(s)
  $w^{n+1} \leftarrow w^n + \beta\, \mathbf{d}^w$ // *Weight updates*
  Update regular triangulation
  $n \leftarrow n + 1$
until (convergence criterion met)

Figure 3.3. *Basic pseudocode of our HOT optimization. Step directions are picked as gradient descent or quasi-Newton steps.*

k-simplex σ^k, if $\mathbf{x}_i$ is any of the vertices of σ^k, the (weighted) circumcenter is expressed as

$$\mathbf{c}(\sigma^k) = \mathbf{x}_i + \frac{1}{2k!|\sigma^k|} \sum_{\mathbf{x}_j \in \sigma^k} \left(|\mathbf{x}_i - \mathbf{x}_j|^2 + w_i - w_j \right) \boldsymbol{\sigma}^k_{\hat{j}}, \tag{3.7}$$

where $\boldsymbol{\sigma}^k_{\hat{j}}$ denotes the inward-pointing normal of the face of σ^k opposite to $\mathbf{x}_j$ weighted by the volume of the face. With this general formula, weighted circumcenters are easy to differentiate, both with respect to vertices and weights. Notice that when the weights of σ^k are all equal, one finds the expression for the (Voronoi) circumcenter used in [51]. Armed with this useful identity, we can now formulate the various HOT energies.

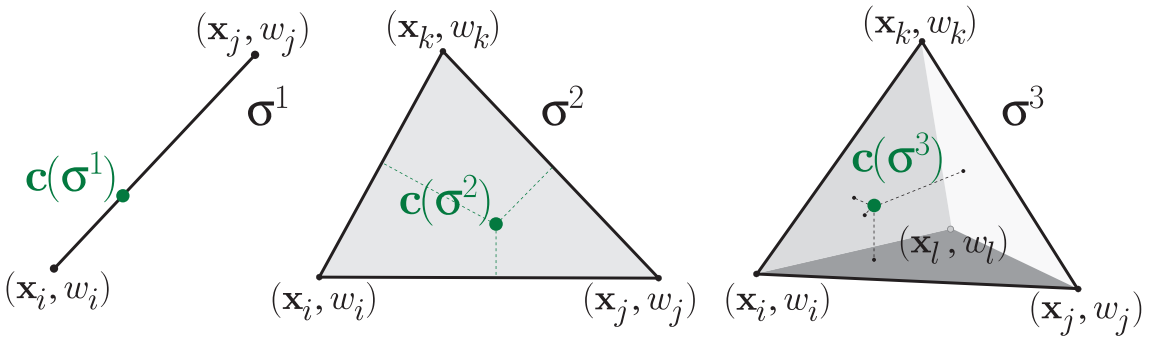


Figure 3.4. ***Weighted Circumcenters****: Weights on primal vertices determine the placement of the weighted circumcenters for each edge (left), triangle (middle), and tetrahedron (right). Moreover, the orthogonal projection of the (weighted) circumcenter* $\mathbf{c}(\sigma)$ *onto one of* σ*'s faces falls on the (weighted) circumcenter of that face.*

3.4.3 $\mathbf{HOT_{2,2}}$ Meshes

When a W_2-based transport cost is used, the HOT functionals are quite easy to compute in closed form. Indeed, a direct application of Pythagoras' theorem reveals that an optimal transport plan to move the normalized uniform measure for a simplex σ to its orthogonal dual $*\sigma$ can be achieved by splitting the plan into two stages: first, optimally transport the measure from σ to its (weighted) circumcenter $\mathbf{c}(\sigma)$, then from $\mathbf{c}(\sigma)$ to the dual cell $*\sigma$. The fact that the circumcenter $\mathbf{c}(\sigma)$ is at the intersection of the mutually-orthogonal affine spaces supporting σ and $*\sigma$ makes the cost of the direct optimal transport and the sum of transport costs via $\mathbf{c}(\sigma)$ equal. The optimal transport cost is thus directly expressible as we now detail.

Energy computations. For both $\star^0$ and $\star^d$ in dimension $d=2,3$, $\mathrm{HOT}_{2,2}$ energies can be easily computed by splitting weighted Voronoi d-cells or primal d-simplices into canonical simplices for which a closed form for the optimal transport cost to a point is easy to obtain—see this splitting in figure 3.5. For instance, for a right triangle T with width a and height b in 2 dimensions, the integral over the triangle of the squared distance to the vertex $\mathbf{p}$ adjacent to a is

$$W(\mathbf{p},T)=\int_0^a\int_0^{\frac{bx}{a}}\left(x^2+y^2\right)dy\,dx=\frac{a^3b}{4}+\frac{ab^3}{12}.$$

The equivalent formula in 3 dimensions for the bi-orthogonal tetrahedron T split shown in figure 3.5 (right) is now:

$$W(\mathbf{p},T)=\int_0^a\int_0^{\frac{bx}{a}}\int_0^{\frac{cy}{b}}\left(x^2+y^2+z^2\right)dzdydx=\frac{abc}{10}\left(a^2+\frac{b^2}{2}+\frac{c^2}{6}\right).$$

Costs for transport from edges are simpler to derive as they only involve 1-dimensional integrals, and the other remaining stars are just combinations of transport over edges, areas, and volumes. For completeness, all the transport costs needed in 2 and 3 dimensions can be found in appendix B.1.

$\star^0$-$\mathbf{HOT_{2,2}}$ in $\mathbb{R}^\mathbf{d}$. As mentioned in section 3.3.3, this energy turns out to be the well-known CVT energy [50] when all the weights are equal. For this particular case, various optimization strategies have been proposed, such as Lloyd iterations (to get near the optimal mesh) followed by a quasi-Newton method (to accelerate convergence) [74], and our approach mimics these strategies in this case. In the general case, however, the gradient with respect to a vertex position does not only contain the usual centroidal term from CVT, but also extra terms based on weight differences. As for the gradient with respect to a weight, its expression is surprisingly simple: it is simply the Laplacian (using the conventional cotangent formula) of the weights, meaning that a $\star^0$-$\mathrm{HOT}_{2,2}$ mesh will have harmonic weights. We can then conclude that a CVT mesh is, in fact, a $\mathrm{HOT}_{2,2}$

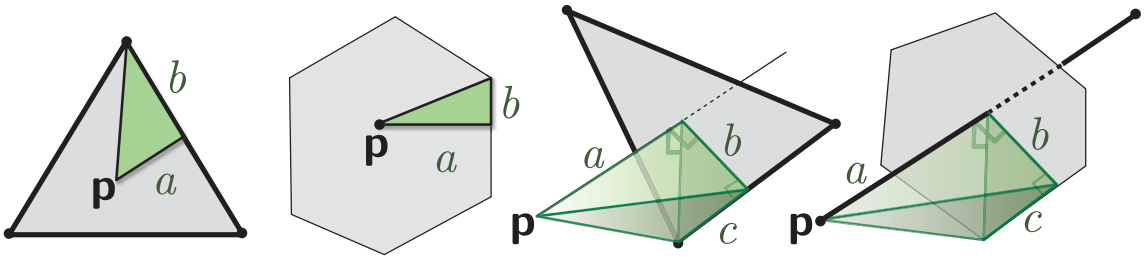


Figure 3.5. ***Splitting Mesh Elements****: Most of our HOT energies are evaluated by splitting simplices/cells into canonical subsimplices (in green) for which closed-form integral expressions $W(\mathbf{p}, T)$ of simplex-T-to-point-$\mathbf{p}$ transport are easily found. Notations used for 2 (left) and 3 (right) dimensions in sections 3.4.3 and 3.4.4 are indicated.*

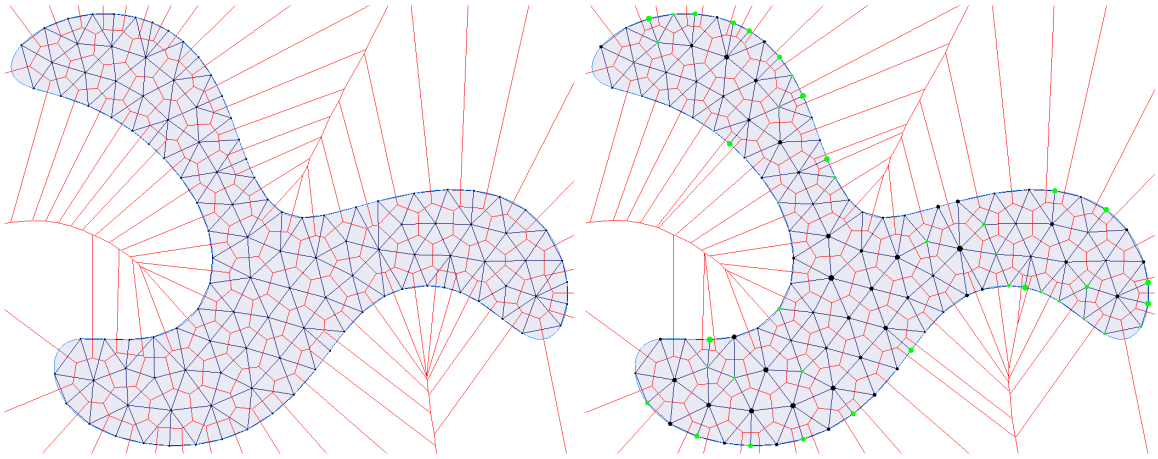


Figure 3.6. ***ODT vs $\star^2$-HOT$_{2,2}$****: An ODT mesh (left) and a $\star^2$-HOT mesh (right) are computed for the same 2-dimensional shape; while the primal triangulations are rather similar, the dual vertices end up closer to the simplices' barycenters for the HOT mesh thanks to the additional freedom provided by the weights.*

mesh with Neumann boundary condition for the weights; other nontrivial boundary conditions will lead to HOT$_{2,2}$ that are not CVT.

$\star^{\mathbf{d}}$-HOT$_{\mathbf{2,2}}$ in $\mathbb{R}^{\mathbf{d}}$. Although seemingly the "dual" version of the $\star^0$ case, this energy requires specific derivations, which we now go over. Computing the energy gradient with respect to weights is made easier if one notices that $\partial \mathbf{c}(\sigma)/\partial w_i$ is independent of the weights: equation (3.7) is indeed linear in the weights of the mesh in any dimension. Consequently, the gradient of the energy (which is quadratic in $\mathbf{c}$) with respect to a weight is linear in the weights of the mesh, offering alternative optimization approaches. One can for instance solve for the optimal weights directly by finding the weights that make all gradients zero: this is simply achieved via a global, sparse linear system collecting all the gradient equations and equating them to zero (see appendix B.1; notice that this system is simply a Poisson problem). A slightly slower approach—but easier to implement and parallelize—is to compute the optimal weight w_i^* for each vertex assuming that the other weights

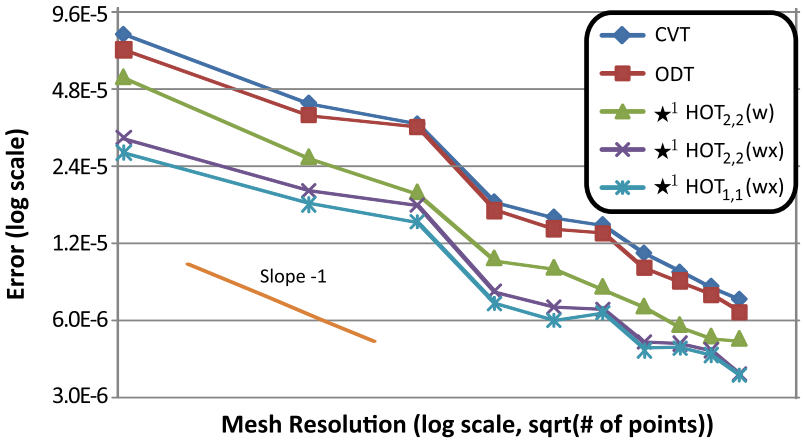


Figure 3.7. ***Laplace equation****: We plot the L^2 errors (with respect to the solution on a very fine mesh) of the solutions of a Laplace equation on a circle with Dirichlet boundary conditions $f(u,v) = (u^2+v^2)\sin(u)\cos(v)$ for CVT, ODT, and HOT meshes with weight optimization only (w) or vertex and weight optimization (wx). While the log-log plot of the errors as a function of mesh size shows, as expected, that the convergence rate is not affected, our HOT meshes still noticeably outperform both CVT and ODT meshes, with W_1 only marginally better than W_2.*

are unchanged, and update $w_i \leftarrow (w_i^* + w_i)/2$ (moving straight to the optimal value may lead to overshooting, hence the half-way update). The gradient with respect to vertices is detailed in appendix B.1, and we optimize vertex positions as sketched in figure 3.3. A $\star^d$-$\mathrm{HOT}_{2,2}$ result in 2 dimensions can be seen in figure 3.6.

$\mathbf{HOT_{2,2}}$ for other stars. Be it in 2 ($d=2$) or 3 ($d=3$) dimensions, the $\mathrm{HOT}_{2,2}$ functionals for $\star^k$ (where $k=1\ldots d-1$) can be derived using the circumcenter formula of equation (3.7). However, they do not simplify in the same way as those for $\star^0$ and $\star^d$ did above. Consequently, a direct application of the general HOT algorithm (figure 3.3) is called for, and it performs as expected.

3.4.4 $\mathrm{HOT}_{1,1}$ Meshes

While the $\mathrm{HOT}_{1,1}$ functionals provide the tightest L_1 bounds on Hodge star errors, their use of the 1-Wasserstein distance makes the equations more difficult: optimal transport plans are often much less obvious to determine, and their costs more difficult to compute. In fact, the $\mathrm{HOT}_{1,1}$ energy for $\star^0$ is directly related to facility location problems [55] that are commonplace in operations research, as it amounts to find the location (sometimes called the (continuous, or integral) geometric median) that minimizes the integrated Euclidean distance to the interior of a polygonal region. Thankfully, closed forms of many of the energies can still be found (albeit, with more difficulty) as detailed in appendix B.2. For a few of the energies, in particular those for $\star^1$ and $\star^2$ in 3 dimensions, closed forms are not easily calculated, and numerical quadrature may be the only practical approach

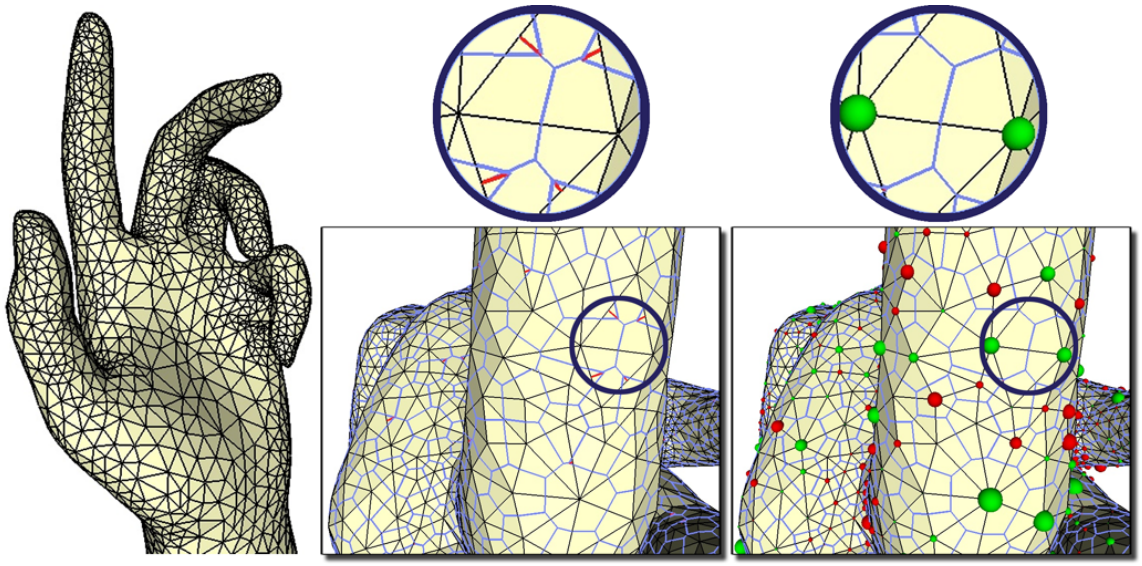


Figure 3.8. ***Surface Weight Optimization****: For a given triangular mesh (left) there are several triangles whose circumcenter is far outside the triangle (center, lines drawn in red). By optimizing only the weights the new dual vertices are better placed inside the unchanged triangles (right) while keeping primal/dual orthogonality.*

for their computation. However, in practice we found the $\mathrm{HOT}_{1,1}$ energies to not give significant improvements over their $\mathrm{HOT}_{2,2}$ counterparts, and thus may only prove useful when the tightest formal bounds are required (figure 3.7).

3.4.5 Discussion

In many ways, HOT meshes can be seen as a generalization of CVT meshes. However, one must be careful with the term "Centroidal Voronoi Tesselation," as being centroidal is a only necessary condition of a CVT energy minimum: for instance, a regular grid is centroidal, and yet the CVT energy is not at a local minimum. Similarly, having each weighted circumcenter at the barycenter of its associated triangle is not sufficient to minimize the $\star^d$-$\mathrm{HOT}_{2,2}$ functional in $\mathbb{R}^d$: the functional also captures the error distribution throughout the domain. A HOT mesh for $\star^k$ tries instead to strike a balance between being "centroidal" or "medial" (i.e., with each k-simplex being "self-centered" for W_p), and having each k cell being of the same volume. In 2 dimensions, most of these energies are globally minimized for a perfect hexagonal tiling of the plane; however, this is no longer true in 3 or more dimensions, as an equilateral simplex no longer tiles $\mathbb{R}^{d>2}$. Consequently, while geometric functionals could be easily derived to simply force a mesh to be centroidal or medial (in the generalized diamond-based sense), HOT functionals also favor uniform sizing of the optimal mesh.

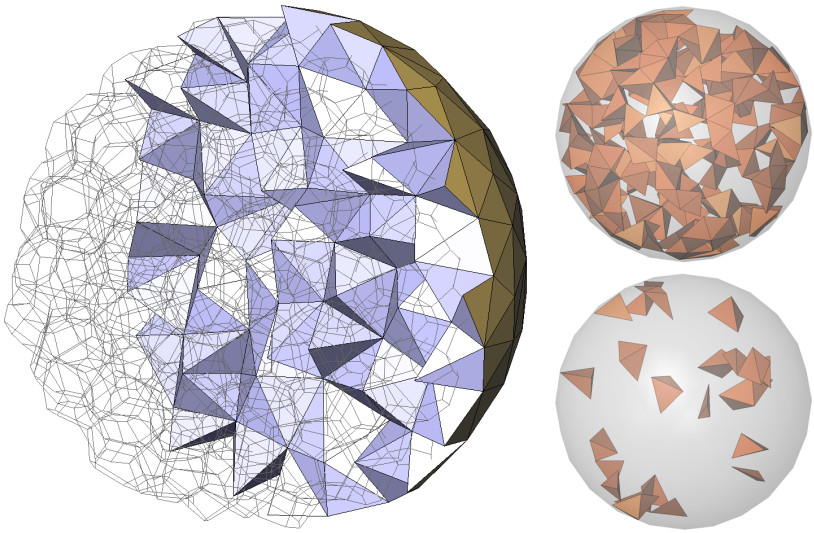
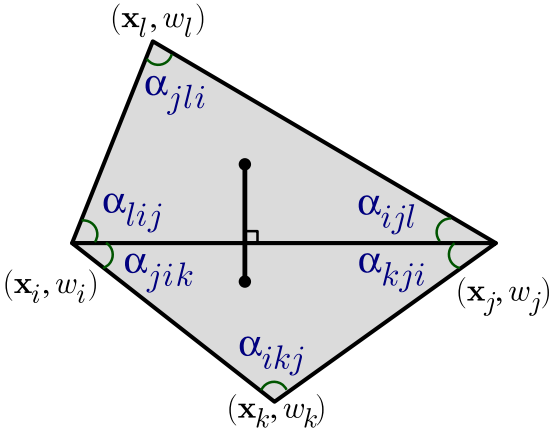


Figure 3.9. ***HOT***$_{2,2}$ ***Sphere****: Optimizing an ODT mesh of a sphere for both weights and vertex positions results in a nice mesh (left) with 30 tetrahedra whose dual vertex is outside of the tet (bottom right), compared to 206 in the original ODT mesh (top right).*

3.5 Applications and Results

HOT meshes can be beneficial in a number of contexts in modeling of surfaces and volumes, as well as in simulation. We mention a few examples to demonstrate the generality of our approach and provide numerical experiments. We also discuss variants and extensions.

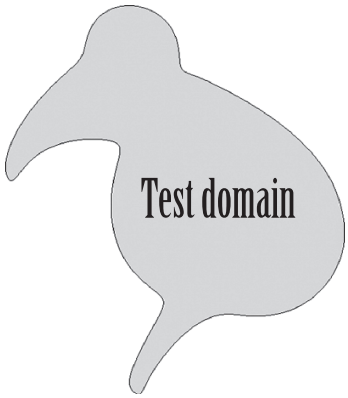
HOT DEC. Most of the DEC methods in graphics (including the huge literature on Laplacian, Laplace-Beltrami, and discrete conformal parameterization) can be directly adapted to work on HOT meshes as well. In fact, if one computes the diagonal Hodge star using the usual ratio of (signed) volume of a simplex and its dual, then no modification is required. If, instead, closed form formulae are used to express each diagonal Hodge star, they only need to be modified to include the contribution due to the weights. For instance, the traditional Hodge star $(\star^1)_{ij} = \big(\cot(\alpha_{ikj}) + \cot(\alpha_{jli})\big)/2$ for a one-form between vertex i and vertex j becomes



$$
\begin{aligned}
(\star^1)_{ij} = \frac{1}{2}\Big(& \cot\alpha_{ikj} + \cot\alpha_{jli} \\
& + (w_i - w_k)\frac{\cot\alpha_{kji}}{||\mathbf{x}_i - \mathbf{x}_j||^2} + (w_j - w_k)\frac{\cot\alpha_{jik}}{||\mathbf{x}_i - \mathbf{x}_j||^2} \\
& + (w_i - w_l)\frac{\cot\alpha_{ijl}}{||\mathbf{x}_i - \mathbf{x}_j||^2} + (w_j - w_l)\frac{\cot\alpha_{lij}}{||\mathbf{x}_i - \mathbf{x}_j||^2}\Big).
\end{aligned}
$$

These changes can be accommodated seamlessly in existing codes, and allow for much greater flexibility: weights can be, for instance, optimized (with fixed connectivity or not) to locally "displace" dual vertices onto an immersed boundary [53] through a least-square fit. Vertices can be optimized as well, for instance in applications requiring local remeshing to maintain good numerics.

Laplace and Laplace-Beltrami operators. A particularly common linear operator in mesh processing is the Laplacian Δ, be it in the plane or on a discrete surface. Its DEC expression for 0-forms being $\Delta = d_0^t \star^1 d_0$ and the d_0 operator being exact, the only loss of accuracy rises from the Hodge star. Consequently, meshes minimizing the HOT energy for $\star^1$ should be appropriate for its accurate computation, as evidenced by figure 3.7 where up to 65% error reduction is achieved compared to CVT. In fact, [68] and [81] were the first to recognize the importance of orthogonal primal/dual meshes to ensure good numerical qualities of the Laplacian. A $\star^1$-HOT$_{2,2}$ mesh indeed results, on a 200V discretization of the test domain depicted in the inset, in a 5% reduction of the condition number of the Laplacian matrix with Dirichlet boundary conditions compared to a CVT mesh (much greater improvements are witnessed when compared to arbitrary, nonoptimized meshes). The result is much more dramatic for the Laplacian of *dual* 0-forms, where the condition number drops from 254 to 90 on the same example. This is partially due to an increase of the minimum dual edge length (going from $2.0e^{-3}$ for CVT to $1.5e^{-2}$ on the same mesh), providing an alternative approach to removing short dual edges presented recently in [79]. Similar improvements were found for the Laplace-Beltrami operator of the surface mesh in figure 3.8.



Improving dual structure. We often have to deal with situations where the triangulation is given and cannot safely be altered. For instance, moving vertices and/or changing the connectivity of a triangle mesh in $\mathbb{R}^3$ is potentially harmful, as it affects the surface shape. Still, the ability to optimize weights to drive the selection of the dual mesh is very useful. We can easily find the weights to, e.g., minimize the L_2 distance squared between weighted circumcenters (defined in equation (3.7) through an equation that is linear in the weights) and triangle barycenters using a single linear solve. The connectivity is kept intact, regardless of the weights—only the position and shape of the compatible dual $\mathcal{D}$ is optimized. Although one cannot guarantee that the resulting dual will be flawless (self-centered and non-self-intersecting), it will be improved compared to the original circumcentric dual. Even for HOT energies, our 2-dimensional and 3-dimensional tests show that only optimizing the weights is particularly simple and beneficial on a number of meshes. Figure 3.8 depicts a triangle mesh of a hand and its intrinsic dual before and after weight $\star^2$-optimization, showing a drastic reduction in the number of negative dual edges—thus providing a practical alternative to the use of

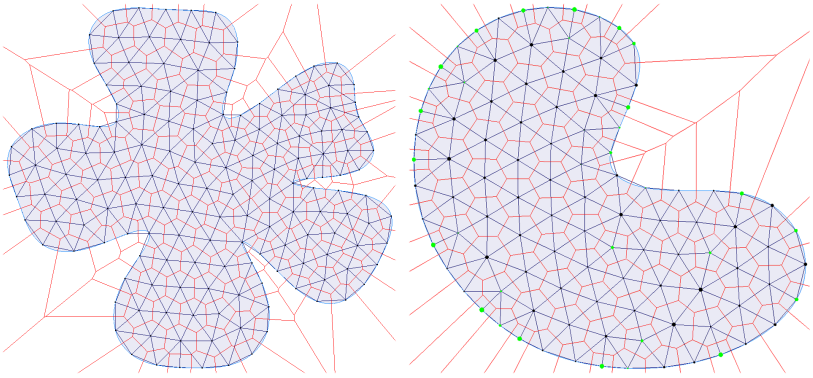


Figure 3.10. ***HOT***$_{1,1}$ ***Meshes****: A "Medial Voronoi Tesselation" (i.e., a* $\star^0$*-*$HOT_{1,1}$ *mesh) has vertices near the integral geometric median of each Voronoi cell (left);* $\star^1$*-*$HOT_{1,1}$ *mesh tends to have primal and dual edges intersecting near their midpoints (right, weights shown as balls with color/size indicating sign/magnitude).*

intrinsic Delaunay meshes advocated in [66]. Similarly, figure 3.11 shows that even an ODT mesh with exceptionally high-quality tetrahedra [78] can be made significantly better centered using a simple weight optimization. Note also that in this example the number of tetrahedra with a dual vertex outside of the primal tet dropped from 17041 on the ODT mesh to 5489 on the HOT mesh—a two-third reduction of "outcentered" tetrahedra. As a final illustrative example we show results on a 3-dimensional sphere (figure 3.9). Starting from an ODT mesh and optimizing only the weights drops the number of outcentered tetrahedra from 206 to 52, while allowing the optimization to also move the positions of the vertices further reduces this number to 30, resulting in the mesh shown in the figure. On the other hand, if a weighted Delaunay mesh is undesirable, optimizing only the positions still reduces the number of outcentered tetrahedra to 118, almost half of the original ODT mesh, while still using a circumcentric (Voronoi) dual.

Accuracy and extensions. While we described archetypical primal-dual HOT energies, one can use regular triangulations and power diagrams to derive other relevant energies. Even in the context of Hodge star accuracy, we point out that the "diamond weighting" proposed in section 3.3 can be modified if one wishes to improve a *particular* Hodge star (and *not* its inverse): for instance, the discrete Hodge star between 0-forms and d-forms in $\mathbb{R}^d$ should use a weighting equal to 1, while the inverse Hodge star should use the volume of the local d-cells. Similarly, one may minimize a linear combination of HOT energies if multiple Hodge stars need to be optimized simultaneously. Designing new energies based on targeted numerical tasks should be straightforward—although continuity and convexity of these functionals will need to be studied on a case-by-case basis. Nevertheless, our $\star^k$-HOT energies lead consistently to a 5% to 35% L^1- and L^∞- improvement on both $\star^k$ and $(\star^k)^{-1}$ for linear and nonlinear functions alike on 2-dimensional nonconvex domains like depicted in the inset earlier—even if the error minimization is not run to convergence. As for the 3-dimensional

Bimba mesh in figure 3.11, our $\star^3$-optimization of only the weights already reduces both the L^1 and L^2 norm of $\star^3$-errors for linear functions by 16%.

$\mathbf{HOT_{1,1}}$ vs $\mathbf{HOT_{2,2}}$. While slower to converge when the 1-Wasserstein distance is used, $\mathrm{HOT}_{1,1}$ and $\mathrm{HOT}_{2,2}$ meshes are visually quite similar. Numerical tests, similarly, do not demonstrate major differences: a simple Laplace's equation with Dirichlet boundary conditions on various mesh sizes clearly indicate that $\mathrm{HOT}_{1,1}$ are slightly better than $\mathrm{HOT}_{2,2}$, but both are significantly better than CVT or ODT (see figure 3.7). $\mathrm{HOT}_{1,1}$ meshes can, in fact, be slightly worse than their W_2 equivalents when their accuracy is tested using polynomial test functions. It is therefore unclear that using the W_1 cost is worth the added computational burden for graphics applications, despite offering tighter theoretical bounds.

3.6 Future Work

Several future directions are ripe for exploration. For instance, formulating other functionals based on particular numerical tasks (such as eigenvalue problems) or other families of functions (other than just Lipschitz) could be of interest. In fact, the induced symmetries of our HOT meshes may improve other operators as well. Deriving L^p-based functionals (using the quadratures pointed in [75]) or incorporating a sizing field in the functionals should be relatively straightforward. We also believe that a sustained effort to produce better optimizations for HOT-like energies is in order to ensure efficient, industrial-strength implementation. Finally, as always in meshing, providing a richer set of boundary conditions would also extend the number of potential applications, thus helping the adoption of HOT meshes. Combining HOT optimization with feature protection through boundary weights as proposed in [80] could offer a practical extension of our approach in this direction.

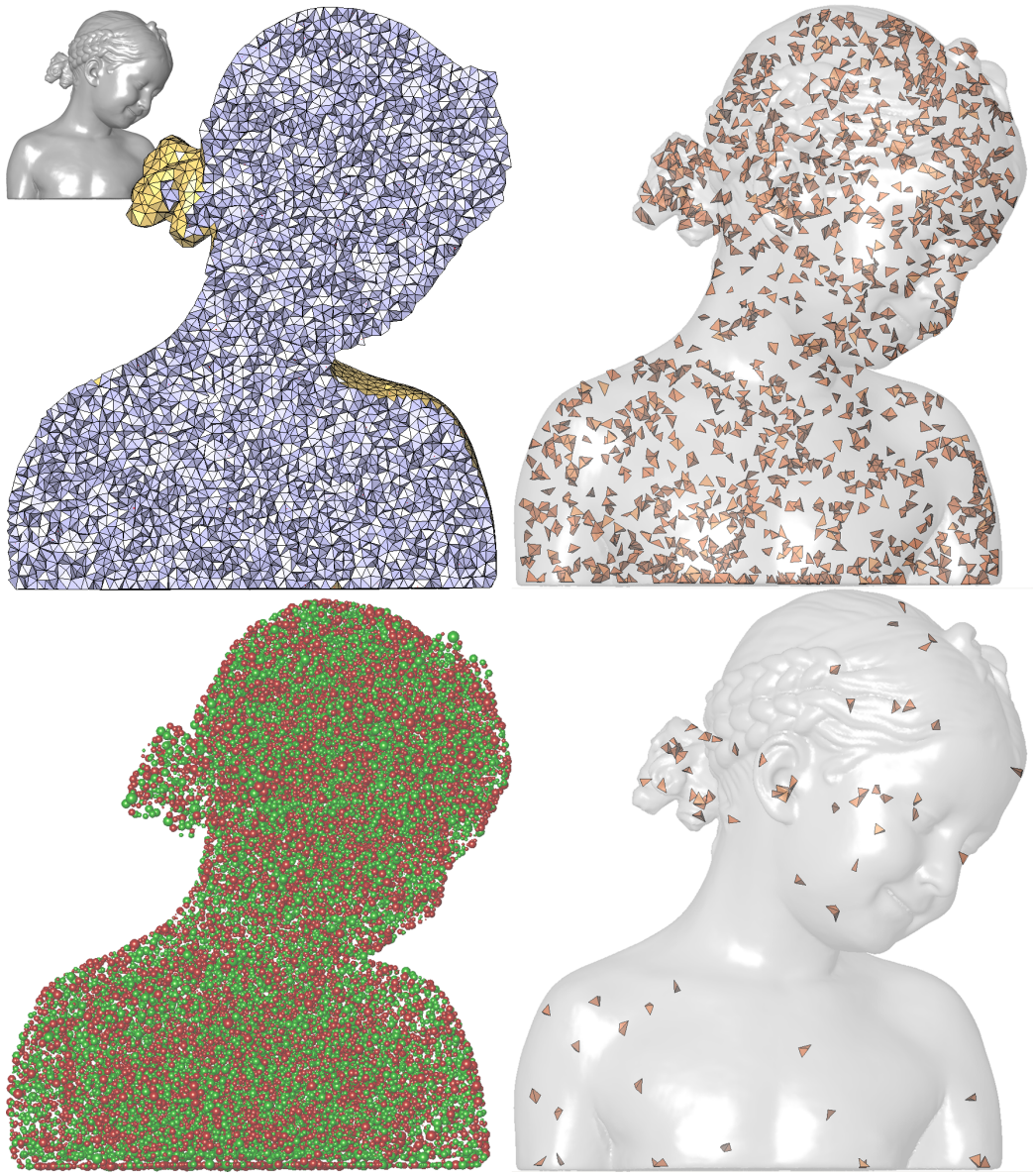


Figure 3.11. ***Weight Optimization in 3 Dimensions****: A high-quality ODT mesh of the Bimba con Nastrino (top left cross section; 195K tets, 36K vertices) can be $\star^3$-optimized by a few (30) iterations of our weight optimization, thus improving minimal dual edge length and self-centeredness (bottom left; weights are displayed according to sign (red/green) and magnitude (radius)). When we single out the tetrahedra with a distance between weighted circumcenter and barycenter greater than 0.5% of the bounding box, one can see the HOT mesh (bottom right) is significantly better than the original ODT (top right), even if the primal triangulations are exactly matching. If we further increase the visualization threshold to the point when the HOT mesh has a single "bad" tetrahedron, the nonweighted original Bimba mesh then exhibits 192 such tetrahedra.*

Chapter 4

Geometric Eulerian Integrators for Fluid Simulation

4.1 Introduction

Having seen in the last chapter how to develop simplicial meshes computationally appropriate for DEC, we now address a problem where complex boundaries often necessitate the use of such boundary-conforming meshes, namely fluid simulation. Physically based animation of fluids is often modeled using the *incompressible Navier Stokes equations*. Numerically integrating these equations presents numerous practical challenges, however, and has been a focus of Computational Fluid Dynamics (CFD) for the past thirty years. When visual impact matters most, more elaborate CFD methods are generally considered unnecessary: simple Eulerian discretizations with explicit semi-Lagrangian or upwind advection have been the methods of choice in computer animation for the last few years.

A significant numerical difficulty in all CFD techniques is avoiding *numerical viscosity*, which gives the illusion that a simulated fluid is *more* viscous than intended. It is widely recognized that numerical viscosity has substantial visual consequences, hence several mechanisms (such as vorticity

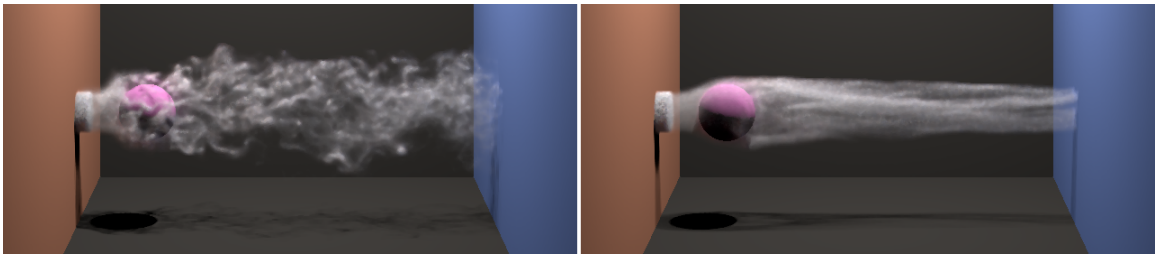


Figure 4.1. *By developing an integration scheme that exhibits zero numerical dissipation, we can achieve more predictable control over viscosity in fluid animation. Dissipation can then be modeled explicitly to taste, allowing for very low (left) or high (right) viscosities.*

confinement and Lagrangian particles) have been devised to help cope with the loss of fine scale details. Common to all such methods, however, is the fact that the amount of energy lost is not purely a function of simulation duration, but also depends on the grid size and number of time steps performed over the course of the simulation. As a result, changing the time step size and/or spatial sampling of a simulation can significantly change its qualitative appearance. Such a dependence can make it difficult to compute coarse previews, for example.

In this chapter, we propose a family of fully Eulerian schemes that provide full control over the amount of dissipation, independent of temporal and spatial resolution. Control over viscosity stems from the use of a nondissipative integration scheme for the Euler equations on arbitrary simplicial grids. We can then add implicit diffusion to model viscosity directly. We demonstrate the efficiency and robustness of these schemes via numerical comparison with current techniques.

4.1.1 Previous Work

Early work on fluid animation for computer graphics widely favored the use of Eulerian staggered grid discretizations [82, 83], where the fluid velocity components are stored per face instead of being collocated at nodes. Semi-Lagrangian advection techniques [84] have been prevalent due to their unconditional stability and ease of implementation. Their resulting excessive energy dissipation was later mitigated by the adoption of vorticity confinement to partially reinject lost vorticity into the flow [85, 83], or through higher-order advection schemes using repeated semi-Lagrangian steps [86, 87]. However, the stability of these methods is guaranteed only if spurious extrema are eliminated by a limiter. More recently, a case was made for explicit, third-order upwind-based advection [88] as a low dissipation technique at reasonable computational cost. Subscale modeling was also proposed [89] to get energy cascading in line with the empirical behavior of statistically stationary isotropic turbulence despite numerical viscosity, although at a scale much larger than intended.

Another remedy to combat dissipation is to add Lagrangian machinery to the Eulerian solver. Selle *et al.* [90] proposed to add vortex particles to track vorticity and inject a tuneable confinement force into the flow. Zhu and Bridson [91] even advocated the substitution of the semi-Lagrangian advection scheme in stable fluids with the Lagrangian *fluid-implicit-particle* (FLIP) scheme [92], which exhibits significantly less numerical dissipation than semi-Lagrangian advection after pressure projection. While fast, these mixed schemes require careful management of particle distribution in order to achieve good quality and performance.

Another line of research in computer animation of fluids focused on offering more versatile spatial discretizations to capture complex geometries with low node count. Spatially adaptive discretizations such as octrees were proposed to improve resolution of highly turbulent flows [93, 94], albeit with significant computational overhead and grid-aligned aliasing artifacts. Integration schemes for

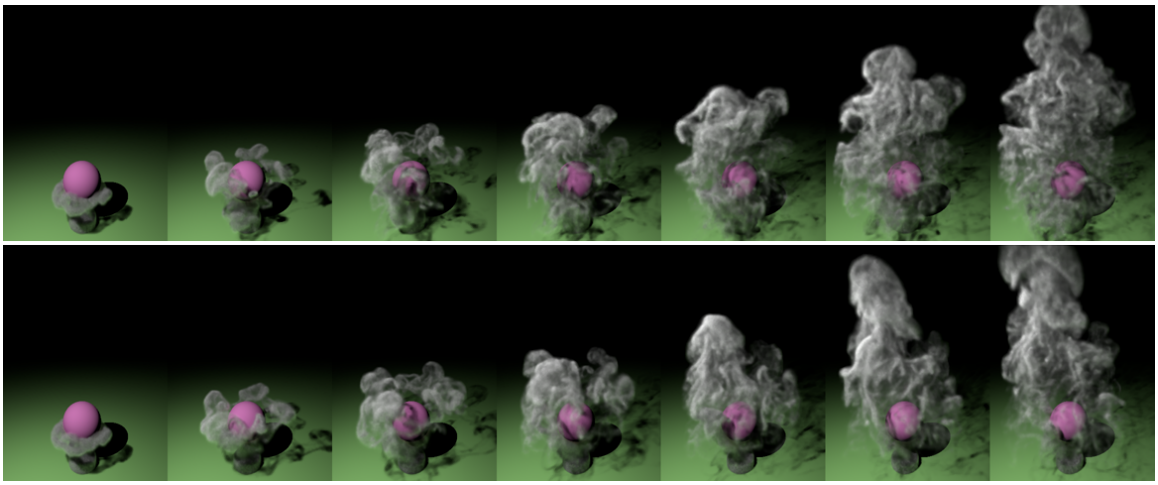


Figure 4.2. ***Flow Past Sphere****: Although completely inviscid flows may look unnatural (top), the absence of numerical viscosity gives animators more predictable control over fluid appearance (bottom); here, smoke rises in a closed box containing a round obstacle.*

simplicial and hybrid meshes through semi-Lagrangian backtracking were introduced, for which the computational overhead brought by non-regular mesh data structures were largely compensated for by the increased visual complexity per element [95, 96]. Even if circulation preservation can be enforced [97], these schemes on arbitrary grids still suffer from noticeable energy dissipation, leading to an uncontrolled energy decay in the flow that depends on both the mesh size and the time step size.

Energy preservation for inviscid fluids on unstructured grids had not received much attention in CFD in the past due to the prevalence of methods based on regular grids. This has changed recently with the introduction of discrete energy-conserving schemes on 2-dimensional and 3-dimensional unstructured grids [98, 99, 100]. The role of discrete differential operators for the curl and divergence (as used in [97]) in ensuring kinetic energy conservation was acknowledged, and numerical benefits from non-dissipative advection were demonstrated. In particular, energy preservation guarantees that the velocity will always remain bounded, bringing unconditional stability to the schemes. Derivations of these new schemes were obtained through discretization of the vorticity form of the continuous Euler equations, but the final update rules require a fairly large stencil to compute advection. Moreover, it is not known whether these schemes can be derived from first principles (as was successfully done for Lagrangian integrators; see [101] for a review).

4.2 Discrete Setup of Fluid Motion

Before describing our Eulerian schemes for fluid animation, we establish the equations of motion of both viscid and inviscid fluids, and discuss the physical properties that our discrete time integrators

should preserve.

4.2.1 Equations of Motion

Consider an inviscid, incompressible, and homogeneous fluid on domain M with velocity $\mathbf{u}$ and pressure p. Assuming constant unit density ($\rho = 1$), the motion of such a fluid is governed by the *Euler equations*, which consist of a *momentum equation*,

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla p, \tag{4.1}$$

along with an *incompressibility constraint*,

$$\nabla \cdot \mathbf{u} = 0. \tag{4.2}$$

Along with boundary conditions, these equations define the fluid behavior and have been thoroughly studied both from a mechanics point of view [102] and from a computational perspective [103].

Vorticity formulation. A particularly convenient expression in which the Euler equations can be rewritten is a function not only of the velocity $\mathbf{u}$, but also of its vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \boldsymbol{\omega} \times \mathbf{u} = -\nabla P, \\ \qquad\qquad \nabla \cdot \mathbf{u} = 0, \end{cases} \tag{4.3}$$

where now P is the Bernoulli pressure, i.e., the former pressure p plus the kinetic energy density: $P = p + \mathbf{u}^2/2$.

Handling viscosity. For a viscous fluid, the incompressible Navier-Stokes equations are used instead. A diffusion term $\nu\Delta\mathbf{u}$ is added to the previous momentum equation as an extra body force, where ν controls the amount of viscosity in the flow. This term diffuses the momentum of the fluid, damping down turbulences.

4.2.2 Relevant Continuous Properties

Euler and Navier-Stokes flows have a number of properties that one may want to preserve in the discrete realm to ensure a close visual match between the discrete results and typical flows in nature. Of particular visual significance is the fact that the Euler equations preserve kinetic energy in time. *Energy preservation* is, however, only rarely observed in practice by the fluid simulators used in computer animation, visually producing overly viscous animation. Additionally, this energy tends to seep toward small scales, a general mechanical property called *energy cascading*: the kinetic

energy is thus conservatively transferred to smaller and smaller length scales statistically, meaning that a large vortex will, over time, erode into several smaller ones.[1] Faithfully reproducing this energy cascading in the discrete case is impossible due to the limited resolution of meshes, and can only be approximated through subscale modeling [89]—although it is quite unclear that the type and scale of animations used in computer graphics can visually benefit from the restricted physical assumptions used in the modeling of forward cascading. Kelvin's circulation theorem (stating that *vorticity is advected along the flow*) was also pointed out in [97] as a crucial property to preserve to visually capture the traditional turbulent behavior of nearly inviscid flows. A last property of the Euler equations that has been shown important to preserve independently of the numerical approximation is *time reversibility* [104]: applying the integration with negated fluxes should exactly run the simulation backwards in time. Notice that this symmetry in time is what was sought after in the BFECC [86] and modified MacCormack [87] schemes: both schemes try to get the backward *and* forward semi-Lagrangian to numerically agree, leading to much decreased numerical viscosity. Finally, some applications are interested in the preservation of the total *enstrophy* of the fluid. However, the benefit of this to computer animation is currently less evident and our methods do not address the preservation of this quantity.

We will thus focus on developing numerical schemes that, by pushing vorticity forward with the flow, will exactly achieve time reversibility and remove numerical energy dissipation for inviscid flows. As we generally want to animate viscous flows, proper treatment of the nonviscous part of the Navier-Stokes equations will ensure proper energy decay even for large time steps. Note that potential artifacts due to energy accumulating at fine scales will be removed with an appropriate viscosity coefficient ν.

4.2.3 Discretization of Physical Quantities

To derive a computational procedure to integrate Euler and/or Navier-Stokes equations, we must discretize both space and time, and define a discretization of our physical quantities as well.

Spatial discretization. We use a tetrahedral mesh $\mathcal{T}$ to discretize the spatial domain M; i.e., the domain is decomposed into a series of tets $\{T_i\}, i = 1...|\mathcal{T}|$. For simplicity of presentation we will assume the mesh is Delaunay and use the Voronoi dual, although the methods are easily applied to regular triangulations and their dual power diagrams as discussed in chapter 3, as the primal/dual orthogonality is the only required property due to our use of the diagonal Hodge star. We assume that these tets are all oriented so as to have positive volume, and their faces and edges are given arbitrary orientations. We denote the Voronoi dual of the mesh as $\mathcal{V}$, having a dual vertex c_i as

[1] Note that the energy may sometimes go back up to larger scales temporarily, as when two same-sign vortices merge.

the circumcenter of tet T_i, dual edge h_{ij} as the edge between c_i and c_j (that is, dual to face f_{ij}), and dual face s_e dual to primal edge e. Note that the dual cells are also given orientations. We will also use the intersections of the primal and dual elements, including c_{ij} (the circumcenter of face f_{ij}) and c_e (the midpoint of edge e). We will denote by $w_{e,j}$ the quad-shaped intersection of cell j and the dual face s_e (see figure 4.3). Finally, $|.|$ will denote the Lebesgue measure (length, area, or volume) of the elements (edges, faces, or tets).

Field discretization. As in [105, 97], we adopt a flux based discretization of the velocity field. This particular discrete setup is well-known, on regular grids, for offering less aliasing than the node-based discretization, as well as for preventing spurious modes in Poisson problems. We will denote by F_{ij} the flux of the fluid through the face f_{ij} common to tet T_i and tet T_j (note that $F_{ji}=-F_{ij}$), while the discrete Bernouilli pressure on the dual node of tet T_i will be denoted P_i. The velocity field will thus be represented by all the fluxes F_{ij}, stored in a vector F of size equal to the number of faces. Similarly, P will represent the pressure field as a vector of all tet-based pressure values.

Discrete operators. We will utilize the DEC operators discussed in section 1.2. Recall that in this notation the **divergence** of $\mathbf{u}$ is directly $\nabla \cdot \mathbf{u} \equiv d_2 F$, i.e., the divergence per tet T_i is the sum of the outward fluxes of T_i . Similarly, the **curl** of $\mathbf{u}$ is represented via its surface integrals over dual faces by $\nabla \times \mathbf{u} \equiv d_1^t \star_2 F$, i.e., the curl per dual face ω_e is the sum of line integrals of $\mathbf{u}$ along the dual edges between adjacent tets around an edge e.

4.3 Discrete Energy-Preserving Time Integrator

In this section, we first present a time integration that preserves total energy while respecting time-symmetry for inviscid fluids, before describing our treatment of viscosity.

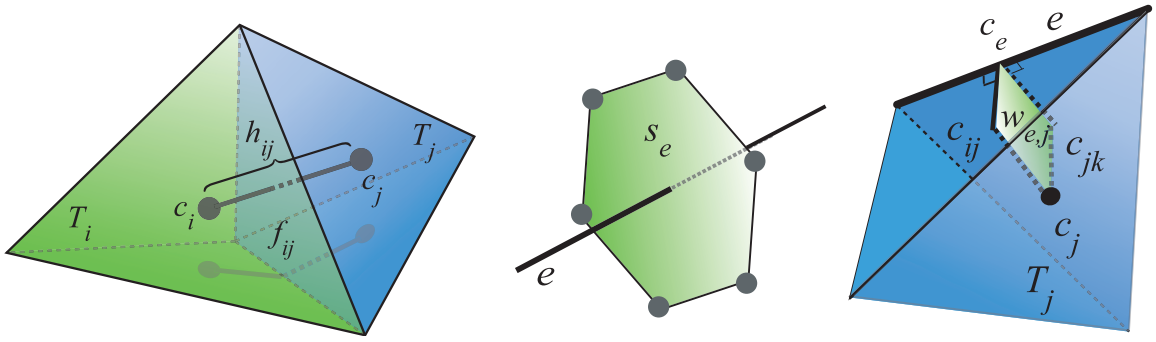


Figure 4.3. ***Spatial Discretization****: we will refer to primal and dual mesh elements using the notation depicted in this figure.*

4.3.1 Discrete Euler Equations

From the vorticity form of Euler equations given in equation (4.3), integrating the continuous terms over each face f_{ij} leads to

$$\dot{F}_{ij} + \overbrace{\int_{f_{ij}} (\boldsymbol{\omega} \times \mathbf{u}) \cdot \mathbf{n}\, dA}^{\mathrm{Adv}(F)_{ij}} = -(\star_2)^{-1}_{ij}(P_j - P_i)$$

$$\sum_{j \in \mathcal{N}(i)} F_{ij} = 0,$$

where $\mathcal{N}(i)$ are the cells sharing a face with i. Note that the diagonal Hodge star $(\star_2)_{ij} = |h_{ij}|/|f_{ij}|$ was used to turn the line integral of ∇P into a surface integral. Using $\mathrm{Adv}(F)_{ij}$ as a shorthand for the area integral over face f_{ij} we just derived, we can write the discrete version of equation (4.3) as a function of F through

$$\begin{cases} \dot{F} + \mathrm{Adv}(F) = - \star_2^{-1} d_2^t P, \\ d_2 F = 0. \end{cases} \tag{4.4}$$

In order to update the Eulerian fluxes F and construct an Euler fluid integrator, we must therefore design a numerical approximation of this "advection term" $\mathrm{Adv}(F)$.

4.3.2 Advection Term

To approximate this advection term, we proceed through local averages of the flux of $\boldsymbol{\omega} \times \mathbf{u}$ on face f_{ij} in a finite volume manner. Our procedure is conceptually simple: first we reconstruct a piecewise-constant velocity vector $\mathbf{U}_i$ per tet T_i based on the known fluxes of each of its four faces: since $\mathbf{U}$ is divergence free there is a unique vector such that the four fluxes of the tet represent the area integral of this vector along the face normals. (Note that this vector can similarly be found through discrete Whitney basis functions to reconstruct the vector field [97].) For each reconstructed vector $\mathbf{U}_i$, we can evaluate the face integral that defines $\mathrm{Adv}(F)_{ij}$: treating the vorticity vector around edge e as a constant $\boldsymbol{\omega}_e = \omega_e/|s_e|.(\mathbf{e}/|\mathbf{e}|)$ in the region, we can integrate the flux of $\boldsymbol{\omega}_e \times \mathbf{U}_i$ over the region of f_{ij} inside the convex hull of the edge e and the circumcenter c_{ij} of the face. Summing the contributions from the two other edges of f_{ij} will provide us directly with the area integral. Finally, we antisymmetrize this evaluation to enforce that our discrete approximation of $\boldsymbol{\omega} \times \mathbf{u}$ also satisfies $\int_M (\boldsymbol{\omega} \times \mathbf{u}) \times \mathbf{u} = 0$; we will see that this will enforce energy preservation when we integrate our equations in time. This procedure provides a low-order approximation of the advection term $\mathrm{Adv}(F)_{ij}$ on each f_{ij}. The reader can find the exact terms involved in this summation, as well as the explanations leading to energy preservation, in appendix C.

Assembly per edge. While this integration can be performed literally as explained above, we found it more efficient to reorganize the terms involved so as to ensure an easy implementation and an efficient evaluation. We instead proceed as follows:

For each edge e,

- evaluate the vorticity ω_e on edge e
- for every pair of consecutive faces f_{ij} and f_{jk} around e,
  - add contribution of F_{jk} and $\boldsymbol{\omega}_e$ on $\mathrm{Adv}(F)_{ij}$:

$$\mathrm{Adv}(F)_{ij} \mathrel{-}= \frac{|f_{ij}|}{h_{ij}} \frac{\omega_e}{|s_e|} F_{jk} \frac{2|w_{e,j}|}{3|T_j|}$$

  - add contribution of F_{ij} and $\boldsymbol{\omega}_e$ on $\mathrm{Adv}(F)_{jk}$:

$$\mathrm{Adv}(F)_{jk} \mathrel{+}= \frac{|f_{jk}|}{h_{jk}} \frac{\omega_e}{|s_e|} F_{ij} \frac{2|w_{e,j}|}{3|T_j|}$$

Note that Adv represents a flux change through oriented faces, thus $\mathrm{Adv}(F)_{ji} = -\mathrm{Adv}(F)_{ij}$; also, consecutive faces must be ordered consistently with the edge orientation and vorticity computation. Now that we have defined the advection operator, we can proceed to build Eqs. (4.4) required to evolve our fluid forward in time.

4.3.3 Time-Reversible Integration

Until now, we only focused on spatial integration of the Euler equations, and did not discuss which fluxes to use in the advection operator. One could use an explicit time integration by using the fluxes at time t^n in $\mathrm{Adv}(F)_{ij}$ to compute the fluxes at time t^{n+1}. Conversely, an implicit integration could be used instead, assuring much better numerical stability at the price of a greater energy dissipation. As argued in section 4.2.2, we would rather derive a time integration devoid of numerical viscosity to offer better control over the fluid viscosity once diffusion is included, as well as enforcing time reversibility.

Therefore, we use a *midpoint* integration for the updates of flux in time, resulting in the following integrator:

$$\begin{cases} F^{n+1} = F^n - h\,\mathrm{Adv}\left(\dfrac{F^n + F^{n+1}}{2}\right) - h \star_2^{-1} d_2^t P^{n+\frac{1}{2}}, \\ d_2 F^{n+1} = 0, \end{cases} \tag{4.5}$$

where $h = t^{n+1} - t^n$ is the time step size, F refers to the vector of all fluxes F_{ij}, and the superscript n (resp., $n + \frac{1}{2}$) is used to indicate evaluation at time t^n (resp., between time t^n and t^{n+1}).

Nonlinear solve. Evolving the fluid in time thus amounts to finding F^{n+1} and $P^{n+\frac{1}{2}}$ such that the following residual is zero:

$$R(F,P) := \begin{pmatrix} \frac{1}{h}(F - F^n) + \mathrm{Adv}\left(\frac{F^n + F}{2}\right) + \star_2^{-1} d_2^t P \\ d_2 F \end{pmatrix}.$$

This update equation allows us to derive the next set of fluxes F^{n+1} as a function of the current fluxes F^n through a nonlinear solve. Notice however that this equation only involves *linear and quadratic terms* in F^{n+1}, so a simple nonlinear solver using Newton's method is sufficient. We thus repeatedly solve the following Newton steps:

$$\overbrace{\begin{pmatrix} \frac{1}{h}\mathrm{Id} + \frac{\partial \mathrm{Adv}}{\partial F} & \star_2^{-1} d_2^t \\ d_2 & \mathbf{0} \end{pmatrix}}^{J} \begin{pmatrix} \delta F \\ \delta P \end{pmatrix} = -R(F^{n+1}, P^{n+\frac{1}{2}}),$$

until the norm of the residual R is below an accuracy threshold.

Improving solver performance. Directly solving the system $J(\delta F^t, \delta P^t)^t = -R$ during each Newton step can be burdensome since J is asymmetric and needs to be assembled for each step. We take two measures to improve performance. First, we approximate the Jacobian matrix J by omitting any entries of its upper left block not on the diagonal. In practice we have found that taking this approach both reduces the amount of work required to set up each Newton step and greatly increases the sparsity of the system without significantly increasing the required number of steps for convergence. We are then left with a typical *saddle-point* problem:

$$\begin{bmatrix} A & \star_2^{-1} d_2^t \\ d_2 & 0 \end{bmatrix} \begin{bmatrix} \delta F \\ \delta P \end{bmatrix} = - \begin{bmatrix} R_F \\ R_P \end{bmatrix}$$

where A is the sum of Id/h and of the *diagonal part* of the Jacobian matrix $\partial \mathrm{Adv}/\partial F$ containing the derivatives of the advection term with respect to fluxes. Notice in particular that, besides being diagonal, the matrix A has only positive terms for sufficiently small time steps (or sufficiently small velocities); hence in practice, our matrix A is positive definite. Saddle point problems of this form can be efficiently solved using the *Schur complement method* [106]. We can indeed solve the linear system:

$$d_2 A^{-1} \star_2^{-1} d_2^t \, \delta P = -d_2 A^{-1} R_F + R_P, \tag{4.6}$$

for the change of pressure δP, then derive the new flux change through

$$\delta F = A^{-1}\left(-R_F - \star_2^{-1} d_2^t \delta P\right).$$

Notice that the pressure update is solved via a Poisson equation (as $d_2 A^{-1} \star_2^{-1} d_2^t$ is a symmetric Laplacian matrix for the metric induced by $\star_2 A$), for which a preconditioned conjugate gradient is most appropriate; thus, applying the Schur complement amounts to solve a system similar to pressure projection to find δP, followed by a trivial backsubstitution to get δF (as A is diagonal). Overall the work done in each Newton step is thus identical to the work done for a single step of the stable fluids algorithm [84].

Further improvement. If the time step needed for animation is small enough, we can further simplify the non-linear solver by only using the dominant term Id/h in matrix A. Discarding $\partial \mathrm{Adv}/\partial F$ renders the matrix A constant, removing the need to rebuild the matrix and allowing for the precomputation of preconditioners. For reasonably sized systems (in our experience, less than 150K tets when using Matlab) we can even LU factorize the Laplacian matrix $h d_2 \star_2^{-1} d_2^t$ once at the beginning, and efficiently solve each Newton step through the Schur complement method in equation (4.6) by constant-time backsubstitution. This simplification is the method of choice in practice when possible. However, for large time steps with respect to the motion, this approach can fail to converge; we then either revert to the previous method and add to A the diagonal of the Jacobian of the advection term, or simply reduce our time step.

4.3.4 Other Time Integration Schemes

While this implicit, time-reversible energy preserving scheme is the one we used in all our examples, we can easily derive other fully Eulerian integrators with similar numerical properties.

Variational integration. Instead of a midpoint rule, we can maintain time reversibility by choosing a trapezoidal time update for which the momentum equation is expressed as

$$\frac{1}{h}(F^{n+1} - F^n) + \frac{1}{2}\mathrm{Adv}(F^n) + \frac{1}{2}\mathrm{Adv}(F^{n+1}) = - \star_2 \, d_2^t P^{n+\frac{1}{2}}.$$

The implementation of this different integrator is nearly identical to the midpoint case, and will thus not be detailed here. Unlike the previous integration scheme, this time-reversible update rule does not conserve energy exactly. Nevertheless, the energy remains nearly constant, basically oscillating around the initial energy. Such a behavior is typical of symplectic integrators, and in fact, we derived this integrator through first principles. This lengthy derivation will not be shown here, but we refer the reader to section 4.4 for an overview of the geometric mechanical arguments leading to this variational time update.

Hybrid time integration. One can also play around with the basic approach and derive other interesting integrators. The vorticity ω used in our derivation can in fact be evaluated at any point

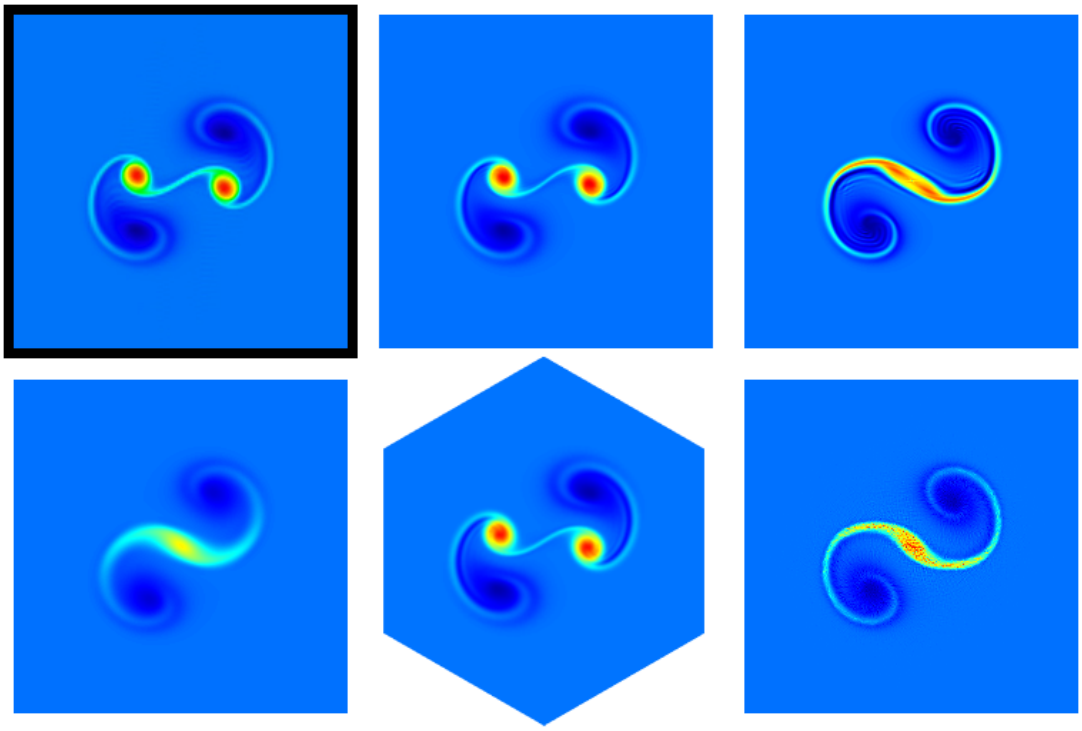


Figure 4.4. ***Taylor vortices on a periodic domain****: for the particular initial separation distance used here, two vortices of the same sign should split apart as in the reference solution (top left). Many schemes fail to reproduce these results. From top to bottom, left to right; Reference solution; Stable fluids [84]; energy-preserving scheme (Harlow-Welch [45] w/ midpoint time discretization); our simplicial energy-preserving scheme; a MacCormack scheme [87]; FLIP [91]. All results were computed on grids of around* 2^{16} *cells or triangles.*

in time between t^n and t^{n+1}, and the update will *remain* exactly energy preserving—while still corresponding to an advection of the vorticity in the velocity field. We can exploit this property to now provide a time update that only requires a *linear* solve in F^{n+1} and $P^{n+\frac{1}{2}}$:

$$\begin{aligned}
&\frac{1}{h}(F^{n+1} - F^n) + \mathrm{Adv}(F^{n+\frac{1}{2}}, \boldsymbol{\omega}^n) = -\star_2^{-1} d_2^t P^{n+\frac{1}{2}}, \\
&d_2 F^{n+1} = 0.
\end{aligned}$$

For a reasonably small time step, this approach offers a fast alternative to midpoint integration. Our experience shows that for larger timesteps, the conditioning of the linear system goes down, and solving this linear system may not always be faster than the full-blown nonlinear alternative. However, this offers a viable fallback solution if the Newton solver of the nonlinear time integrator fails.

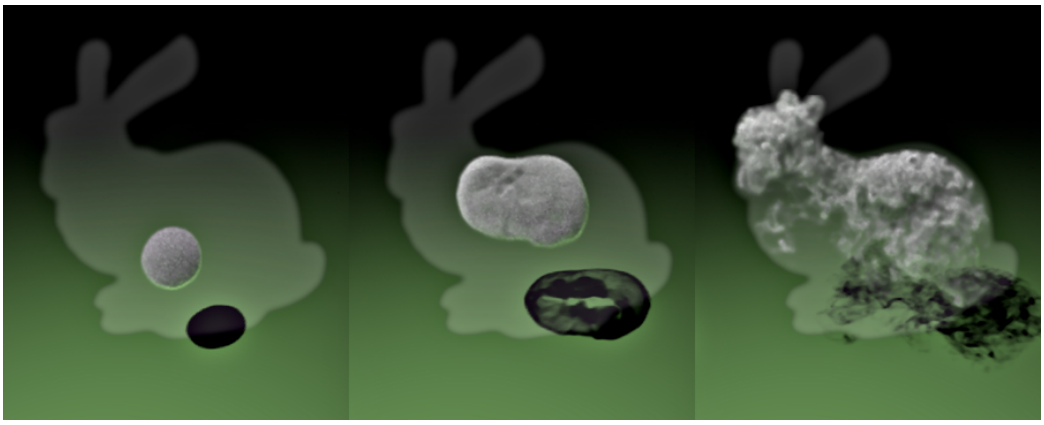


Figure 4.5. ***Smoking Bunny****: hot smoke rising in a bunny-shaped domain, then cooling down over time.*

4.3.5 Viscosity

In practice, fluid animation in computer graphics requires a small amount of viscosity to render the motion more realistic. We thus need to approximate the dissipation term $\nu\Delta\mathbf{u}$ to be added to the right-hand side of equation (4.1) to transform the Euler equations into the Navier-Stokes equations. As we represent the divergence-free velocity field $\mathbf{u}$ by its fluxes on mesh faces, we can directly apply the discrete Laplacian operator as defined in [97], which, with our notations, is expressed as $-d_1 \star_1^{-1} d_1^t \star_2$. Therefore, we modify our integrator to include this term evaluated at the midpoint (we will denote $\bar{F}^{n+\frac{1}{2}} := (F^n + F^{n+1})/2$), resulting in the following momentum update rule:

$$F^{n+1} = F^n - h\left[\mathrm{Adv}\left(\bar{F}^{n+\frac{1}{2}}\right) - \star_2^{-1} d_2^t P^{n+\frac{1}{2}} - \nu d_1 \star_1^{-1} d_1^t \star_2 \bar{F}^{n+\frac{1}{2}}\right], \tag{4.7}$$

This simple dissipation model turns out to be mostly independent of the time step size and the spatial discretization. As we will see in section 4.5, this is particularly convenient for computer animation, as one can easily and predictably adjust the viscosity of the fluid simulated without having to worry that the visual results are, in fact, dependent on the time step. Note that this also allows adaptive time stepping strategies to be used for more efficient computations without inducing motion artifacts.

4.3.6 Boundary Conditions

Basic boundary conditions can be dealt with quite simply in our Eulerian framework. First, we can control the normal component of the velocity along the boundary by specifying the desired fluxes F_{ij} on boundary faces. No-transfer conditions are thus implemented by simply setting fluxes to zero, while forced fluid influx or outflux is achieved by forcing nonzero F_{ij} values in the solver (note

that to ensure divergence-freeness, the total flux through the boundary should still sum to zero). We can also prescribe *pressure* values on the outside of boundary faces to achieve "open" boundary conditions, where the boundary flux will now be determined by the gradient of the pressure across the boundary. Both types of boundary conditions fix the same degrees of freedom for the momentum update rule, although if only flux boundary conditions are used, the pressure is arbitrary up to a constant, in which case the pressure of one tet may be fixed to keep the system definite.

Tangential velocity conditions may, however, seem less obvious since we encoded the fluid velocity only by its normal component to each face: while Harlow and Welch [45] implement free-slip condition by mirroring the tangential velocity component across the wall and no-slip boundary condition by reverting the tangential component across the wall, this type of symmetrization of the velocity is no longer simple for simplicial grids. Fortunately, we can implement the same boundary conditions by simply acting on the vorticity instead. For instance, free-slip condition is achieved by setting the vorticity on boundary edges to 0 since if the half-Voronoi face (loop of dual edges) is completed by its mirror image across the boundary surface, copying the tangential velocity component on this mirror image will cancel out the local vorticity integral. Conversely, no-slip condition is achieved by setting the vorticity on boundary edges to be set to the sum of existing dual circulations $(\star_2)_{ij}F_{ij}$ along the half-Voronoi face: reverting the tangential velocity component on the mirrored half Voronoi face simply double the integral over the full Voronoi face, leaving only the terms including inside fluxes in the vorticity. Partial slip can then be implemented by combining these two and using only some percentage of the vorticity around boundary edges, allowing the modeling of varying "roughness" of boundary materials. Note the choice of the tangential conditions does not affect the energy preservation in the abscence of viscosity.

4.3.7 Discussion

One can easily show that the midpoint integration we introduced is energy preserving: indeed, $\bar{F}^{n+1/2}\star_2 \mathrm{Adv}(\bar{F}^{n+1/2})$ is zero due to the antisymmetrization of the advection term (see appendix C). Therefore, assuming the solver converges, this scheme preserves energy exactly up to machine accuracy and solver tolerance—resulting in stability as the velocity must thus remains bounded. This midpoint integrator on unstructured grids is, in fact, an extension of Harlow and Welch's scheme [45] designed for regular grids: the same reasoning (either through the finite volume derivation presented in this chapter, or via a variational derivation) leads to their skew-symmetric, conservative form of the advection when applied to regular hexahedral cells. Notice that this advection was used by Foster and Metaxas [82], albeit with an explicit integration in time and an iterative pressure projection—both of which lead to a change of energy. Instead, our approach shares most of the well-studied numerical benefits of the original Harlow-Welch conservative scheme [107]. Our integrator also involves a sparser system of equations than the most recent energy-preserving fluid integrators on

unstructured meshes [99, 100], as only the topological one-ring of a face is involved in its update. While this property also means a lower-order accuracy, it provides fast fluid animation at low cost. Further, the precise choice of mesh refinement or time step (which can be nonintuitive to an animator) becomes less critical in the design of an animation, and no extra parameters (like a number of Lagragian particles, or a coefficient of vorticity restitution) is needed. Note finally that simulating fluids in 2 dimensions basically follows the same procedure, with the edge/face/tet volumes replaced by vertex/edge/face volumes, and the 2/3 coefficient in the advection replaced by 1/4.

4.4 Alternate Derivation of Symplectic Integrator

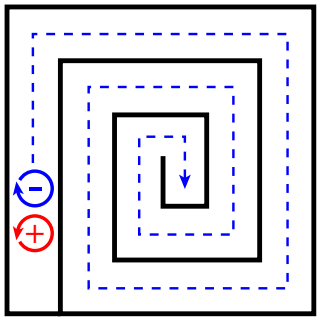
Our trapezoidal-based numerical integrator (section 4.4) can also be derived from first principles. We only sketch the derivation here—a full derivation can be found in [108]. While time integrators for fluids are often derived by approximating equations of motion, we instead discretize the configuration space of incompressible fluids and then derive the equations of motion through the principle of stationary action. Our approach uses an Eulerian representation of *discrete, volume-preserving diffeomorphisms* that encodes the displacement of a fluid from its initial configuration using matrices whose rows and columns sum to one. From this particular discretization of the configuration space, which forms a finite-dimensional Lie group, one can derive a discrete equivalent to the Eulerian velocity through its Lie algebra, i.e., through matrices whose rows and columns sum to zero. After imposing nonholonomic constraints on the velocity field to ensure physical transfer only between neighboring cells during each time update, we apply Lagrange-d'Alembert principle (a variant of Hamilton's principle for nonholonomic systems) to obtain the discrete equations of motion for our fluid representation. The update rule obtained this way (using a simple finite-volume advection operator to define kinematic advection) corresponds to what we detail in section 4.4. The resulting Eulerian variational Lie-group integrator is structure-preserving, and as such, has numerous numerical properties, from momentum preservation (through a discrete Noether theorem) to good long-term energy behavior. This approach also benefits from being easily generalizable to a larger class of problems, including semidirect products as done in [109], which encompasses the behavior of physical systems such as magnetohydrodynamics and complex fluids.

4.5 Results

To validate our approach, we ran a series of tests in 2 and 3 dimensions, for both visual and numerical evaluations. Note that we used Delaunay meshes for our domains, as this prevents having to deal with negative dual edge lengths or dual surface areas. One could locally revert to the barycentric dual as well, most likely at the price of accuracy.

Vortices in periodic domain. A setup commonly used in CFD was used in 2 dimensions to evaluate the quality of the integrators presented in this chapter. A periodic 2-dimensional domain was initialized with two Taylor vortex distributions of same sign for which pseudospectral solutions (with 3/2 dealiasing) are available in the literature. As figure 4.4 shows, our results match the expected qualitative behavior (vortices separating) on any (reasonably good) triangle mesh. With the exception of the FLIP method, which, while failing to capture the proper behavior, still results in visually pleasing results, none of the other methods were found to provide reasonable results on arbitrary grids.

Spiral maze. We also used a 2-dimensional example with many boundaries forming a spiral maze, to demonstrate how much diffusion previous methods engender: while the initial conditions are set up to create a vortex that should propagate through the maze, all our tests of other methods exhibit either significant dissipation, or unexpected behavior. However, our time-reversible scheme does advect the vortex along; no viscosity was added in this example to emphasize energy preservation.



Flows in 3 dimensions. Examples we tried in 3 dimensions include particles blown in a flow past a sphere (figure 4.2), and smoke in a $31K$-tet bunny-shaped domain (figure 4.5; buoyancy is incorporated based on the local density of marker particles passively advected through the flow). These examples were chosen to offer easy comparison with previous methods. In these examples, we set our nonlinear solver l_∞ threshold to 10^{-3}, and the typical range of Newton steps needed to reach convergence was between 3 and 15. We found that in practice our method was not substantially slower than existing schemes for velocity advection, and it is worth pointing out that rendering or advecting a density field in the flow still largely dominates the cost of animation. Therefore, it is well worth a little extra time on velocity integration, especially since it can have a profound effect on the appearance of the final animation. All of our 3-dimensional examples took no more than an average of 40 seconds per time step, and we typically used 2—6 time steps per frame. Additionally, we found that a naive adaptive time stepping method based on the CFL condition helps improve the computational complexity without detriment to the dynamics. A change of time step size in previous methods does, in contrast, significantly alter the results as it introduces a change of viscosity in time.

Numerical comparison of dissipation. As a stress test to estimate numerical energy dissipation of typical fluid integrators used in graphics, we computed the kinetic energy in time for the simulation of an inviscid flow on a periodic domain in figure 4.6. While the decay rate varies significantly from method to method, no integrator but ours is devoid of numerically induced energy dissipation. Notice that even with a very low Newton's accuracy threshold of 10^{-1}, our midpoint implicit Eulerian

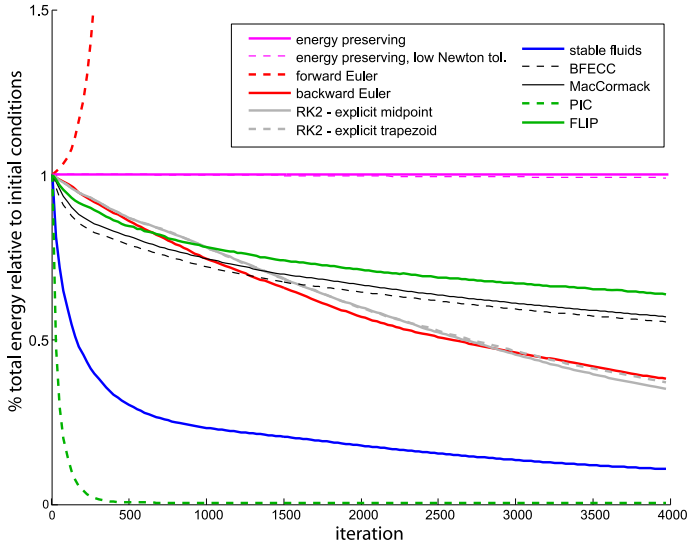


Figure 4.6. *Even for completely inviscid flows, time integration schemes used in computer animation dissipate a significant percentage of total energy over time. Our fully Eulerian scheme not only preserves energy exactly, but also demonstrates excellent energy behavior even for very approximate solutions (here we use an l_∞ tolerance of 10^{-1} in the Newton solver).*

scheme still preserves energy remarkably well. We also tried to adjust the parameters involved in vorticity confinement and vortex-particle-enriched methods to limit energy loss. However, as figure 4.7 indicates, finding good coefficients to eliminate dissipation as much as possible is very animation dependent and creates rather unpredictable energy behavior, making the parameter-tweaking process difficult and unintuitive. In contrast, our approach requires no parameter tuning.

Robustness to time step. We also confirmed the robustness of our implicit integrators with respect to time step and grid sizes in the realistic context of viscous flows. Such improved numerics makes the design of fluid animations easier, as the viscosity parameter will have a predictive value on the results, instead of depending heavily on other simulation parameters (see figure 4.1).

4.6 Conclusion

Fully Eulerian implicit integrators have been largely unexplored in graphics. We have presented evidence that they not only offer a robust computational tool for fluid integration, but possess numerical qualities highly desirable in animation: damping of the fluid flow is no longer a numerical artifact, but a controllable parameter.

Future work. Methods such as FLIP used in [91] effectively eliminate numerical diffusion (before pressure projection) but *not* numerical energy dissipation, as demonstrated in section 4.5. In con-

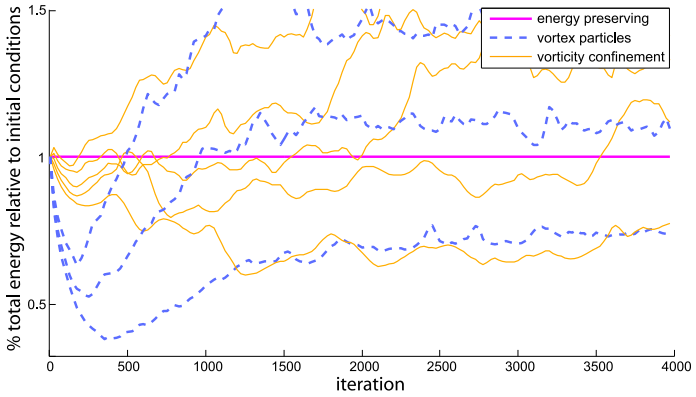


Figure 4.7. *For carefully chosen parameter values, some existing schemes can roughly preserve initial energy; even then, energy behavior is highly unpredictable. We show the energy curves resulting from several nearby parameter values for each scheme above.*

trast, our fully Eulerian approach is devoid of numerical energy dissipation, but does not eliminate diffusion: vorticity diffusion is unavoidable in a purely Eulerian context, since discretization onto a fixed grid acts as a low-pass filter of the velocity field. Particle-based methods do not have this problem since Lagrangian particles carry information instead of diffusing it around. However, the nature of fluid flows makes the Eulerian approach particularly convenient, as no seeding or redistribution of particles is needed even for vastly turbulent flows. It may be worthwhile to develop Eulerian schemes with lower diffusion, while maintaining their energy-preserving characteristics. Another interesting research direction is the design of better subscale modeling models; however, adding noise in the flows may well be more practical for graphics application than subscale modeling. Nonlinear schemes can also be fairly well approximated through less computationally intensive explicit integrators [107], although most likely at the cost of more stringent condition on time step size. Finally, the type of conservative integrators we discussed in this chapter may be very appropriate for coarse-to-fine design of fluid animation, possibly offering an *Eulerian* extension to Lagrangian tracking methods [110].

Chapter 5

Conclusions

In this final chapter, we will first review the contributions of the work presented in the previous chapters, followed by a discussion of the high-level lessons learned and the potential future work they point to.

5.1 Review of Contributions

We first extended the discrete exterior calculus machinery by introducing discretizations of contraction and Lie advection with low numerical diffusion, and important tool previously missing from the DEC toolbox. By directly discretizing Cartan's formula we preserved the fact that a closed form remains closed when Lie advected. This work can also be seen as an extension of classical numerical techniques for hyperbolic conservation laws to handle advection of arbitrary discrete differential forms. In particular, the scheme in 3 dimensions is a generalization of finite volume techniques where not only *cell averages* are used, but also *face* and *edge averages*, as well as *vertex values*. We demonstrated that this generalization preserves the low diffusion behavior of the corresponding finite volume method.

Next we provided a unifying approach to mesh quality based on the placement of primal *and* orthogonal dual elements, while most previous meshing methods focused on designing well-shaped primal triangulations *or* dual complexes. In an effort to provide meshes most appropriate for fast yet reliable DEC-based computations, we proposed functionals on regular triangulations along with their associated power diagrams that offer formal bounds on the numerical error induced by the use of diagonal Hodge stars. Moreover, we unveiled an important connection between Hodge star accuracy and optimal transport. We then demonstrated that meshes that minimize these functionals have desirable geometrical and numerical properties, resulting in Hodge-optimized meshes, which offer a much-needed alternative to the traditional use of barycentric or circumcentric duals in discrete computations. The resulting set of meshing tools has wide applications: even when a specific connectivity is needed, some of our contributions can be applied to improve condition numbers of

basic operators as well as increase numerical robustness and accuracy.

Finally, we presented purely Eulerian integrators for fluid flow on simplicial grids. These integrators involve very sparse equations, are unconditionally stable, and can be made time reversible, while exhibiting excellent long-term energy behavior in the sense that total kinetic energy of inviscid fluids is conserved over arbitrarily long durations without any parameter tuning. In the important case of viscous fluids, our schemes capture the correct energy decay, mostly independently of the time step and grid size. Efficiency was achieved via an application of the Schur complement to quickly solve the saddle-point problem arising from the implicit integration step. Consequently, the implicit schemes are nearly as efficient as other integrators for typical animation sequences, with the benefit of being robust to change of time and space resolution. We showed that the resulting schemes share the same numerical benefits as the Harlow-Welch scheme with a Crank-Nicolson time discretization on regular grids, and discussed how some of these integrators are, in fact, variational, offering a connection between discrete forms and discrete diffeomorphisms that has proven useful in several applications.

5.2 Takeaways and Future Work

As discussed in section 1.1.2, Eulerian methods have both their advantages and disadvantages. While the goal of this thesis was to further the state of the art in geometric Eulerian methods, in practice the disadvantages of purely Eulerian methods remain as evident as their advantages. In particular, when dealing with nonsmooth geometry or dynamics, diffusion can quickly become a problem if care is not taken. For Lie advection on regular grids this problem was dealt with using larger stencils with conditional upwinding at the price of higher computational costs and memory requirements. A similar approach could in theory be used for the incompressible Navier-Stokes equations, but such methods are not nearly as easy on non-regular-grid spatial discretizations, and the conditional nature of such methods can quickly break the antisymmetry required for the energy preservation. At the same time, the nonlinearity of the Euler equations can greatly amplify the computational costs induced by larger stencils or conditionals, quickly turning many approaches impractical. Moving boundaries, be they free surfaces or just time-varying domains, bring up similar issues, and it is clear that more work needs to be done to better address such problems. Likely some hybrid Eulerian/Lagrangian methods will be needed for the practitioner, be they ALE-like, moving the mesh along with the dynamics, or involving the addition of particles or similar elements.

However, how to introduce such extensions in such a way that the desirable geometric properties such as energy preservation or symplecticity are preserved is not clear. For example, a remeshing method that preserves the symplectic form of the dynamics could have enormous benefits. The weighted triangulations discussed in chapter 3 may be able to play a role in this future, as changing

only a scalar per vertex can control the entire orthogonal dual structure without altering the mesh topology, allowing increased control of the location of important quantities like pressure, which becomes crucial in dealing with boundary conditions of free surfaces [53]. By placing quantities on appropriate elements perhaps adapting only the dual mesh would be sufficient to introduce enough "Lagrangianess" to alleviate a large quantity of the diffusion while remaining computationally tractable.

There remain an enormous number of applications ripe for the type of approaches discussed in this thesis that remain to be explored. While some of these ideas have recently been extended to a more general class of theories, including magnetohydrodynamics and complex fluids [109], their application to other areas could be equally fruitful. For example, most methods for discretizing relativity, including the famous Regge calculus [111], fail to satisfy important geometric properties such as the Bianchi identities. In addition, they can operate on simplicial meshes in dimensions higher than 3, and developing appropriate meshing techniques along with the operators is also important. One large step in this direction would be to generalize the notions of DEC to operate on not only differential forms, but arbitrary tensors, in such a way that key geometric identities are preserved.

Finally, while high-resolution methods are useful in preserving sharp features in advection problems, high-order methods, and in particular spectral methods, have proven both powerful and useful in many fields including fluid simulation. Low-order approximations like the diagonal Hodge star are convenient and computationally efficient, but extending the dynamic discretizations presented here to spectral convergence in a computationally efficient manner, as well as developing appropriate meshes for such methods, is an important step that combines the the advantages of fast convergence with the power of geometric discretizations. While there has been some work along these lines for quadrangular grids, for example [112, 113], a full framework remains to be developed.

Appendix A

Details on HOLA-7

For completeness, we provide the stencils and weights for a HOLA-7 advection. Third-order stencils are used, requiring 4 values per stencil. The 4 stencils can be combined using the weights $w_0 = \frac{1}{35}$, $w_1 = \frac{12}{35}$, $w_2 = \frac{18}{35}$, and $w_3 = \frac{4}{35}$ to obtain a 5th-order approximation at a point in smooth regions [114]. The following q_i (see figure A.1) compute the integral of the reconstructed ρ from the boundary between ρ_x and ρ_{x+1} to the boundary minus h for each stencil (generally $v \cdot dt$), where Δx is the width of each grid cell.

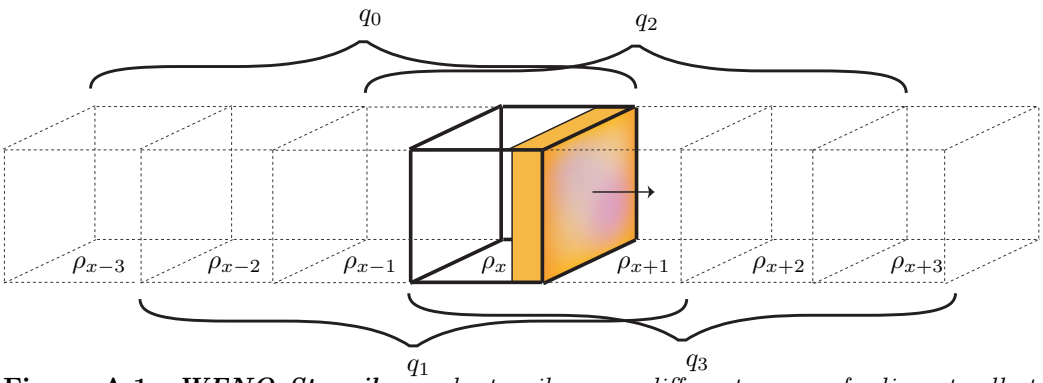


Figure A.1. ***WENO Stencils****: each stencil uses a different group of adjacent cells to compute the integral of a polynomial reconstruction over the orange region. Note all stencils used include the upwind (bold) cell ρ_x.*

$$
\begin{aligned}
q_0 &= \frac{h^4}{24\Delta x^3}(\rho_{x-3} - 3\rho_{x-2} + 3\rho_{x-1} - \rho_x) && + \frac{h^3}{12\Delta x^2}(-3\rho_{x-3} + 11\rho_{x-2} - 13\rho_{x-1} + 5\rho_x) + \\
&\quad \frac{h^2}{24\Delta x}(11\rho_{x-3} - 45\rho_{x-2} + 69\rho_{x-1} - 35\rho_x) && + \frac{h}{12}(-3\rho_{x-3} + 13\rho_{x-2} - 23\rho_{x-1} + 25\rho_x) \\
q_1 &= \frac{h^4}{24\Delta x^3}(\rho_{x-2} - 3\rho_{x-1} + 3\rho_x - \rho_{x+1}) && - \frac{h^3}{12\Delta x^2}(\rho_{x-2} - 5\rho_{x-1} + 7\rho_x - 3\rho_{x+1}) - \\
&\quad \frac{h^2}{24\Delta x}(\rho_{x-2} - 3\rho_{x-1} - 9\rho_x + 11\rho_{x+1}) && + \frac{h}{12}(\rho_{x-2} - 5\rho_{x-1} + 13\rho_x + 3\rho_{x+1}) \\
q_2 &= \frac{h^4}{24\Delta x^3}(\rho_{x-1} - 3\rho_x + 3\rho_{x+1} - \rho_{x+2}) && + \frac{h^3}{12\Delta x^2}(\rho_{x-1} - \rho_x - \rho_{x+1} + \rho_{x+2}) + \\
&\quad \frac{h^2}{24\Delta x}(-\rho_{x-1} + 15\rho_x - 15\rho_{x+1} + \rho_{x+2}) && - \frac{h}{12}(\rho_{x-1} - 7\rho_x - 7\rho_{x+1} + \rho_{x+2}) \\
q_3 &= \frac{h^4}{24\Delta x^3}(\rho_x - 3\rho_{x+1} + 3\rho_{x+2} - \rho_{x+3}) && + \frac{h^3}{12\Delta x^2}(3\rho_x - 7\rho_{x+1} + 5\rho_{x+2} - \rho_{x+3}) +
\end{aligned}
$$

$$\frac{h^2}{24\Delta x}(11\rho_x - 9\rho_{x+1} - 3\rho_{x+2} + \rho_{x+3}) \qquad + \frac{h}{12}(3\rho_x + 13\rho_{x+1} - 5\rho_{x+2} + \rho_{x+3})$$

The smoothness function we used is the sum of the integral squared of each derivative of the reconstructed polynomial over the region of the reconstruction Ω, i.e.,

$$S(\rho(x)) = \sum_{k=1}^{3} \int_{\Omega} (\frac{\partial^k}{\partial x^k}\rho(x))^2 \, dx. \tag{A.1}$$

For the cubic polynomial constructed from ρ_0, ρ_1, ρ_2, and ρ_3 this evaluates to

$$\begin{aligned} S(\rho_0, \rho_1, \rho_2, \rho_3) \quad = \quad & \frac{1}{60}(867\rho_0^2 + 6083\rho_1^2 - 11606\rho_1\rho_2 + \\ & 6083\rho_2^2 + 3802\rho_1\rho_3 - 4362\rho_2\rho_3 + 867\rho_3^2 - \\ & 2\rho_0(2181\rho_1 - 1901\rho_2 + 587\rho_3)). \end{aligned}$$

We also tried several other smoothness functions, including weighting the derivatives based on their order and varying the range of integration, but found little to no qualitative differences in the results. The above equations are all used in standard WENO fashion to compute the final flux through the face between ρ_x and ρ_{x+1} (assuming $h = v \cdot dt$ is positive) as

$$\frac{1}{\alpha}\sum_{i=0}^{3} \frac{w_i q_i}{S(\rho_{x+i-3}, \rho_{x+i-2}, \rho_{x+i-1}, \rho_{x+i}) + \epsilon}, \tag{A.2}$$

where

$$\alpha = \sum_{i=0}^{3} \frac{w_i}{S(\rho_{x+i-3}, \rho_{x+i-2}, \rho_{x+i-1}, \rho_{x+i}) + \epsilon} \tag{A.3}$$

is a normalization factor, and ϵ is a small number to avoid division by 0 (taken as 1×10^{-6} in our experiments).

Appendix B

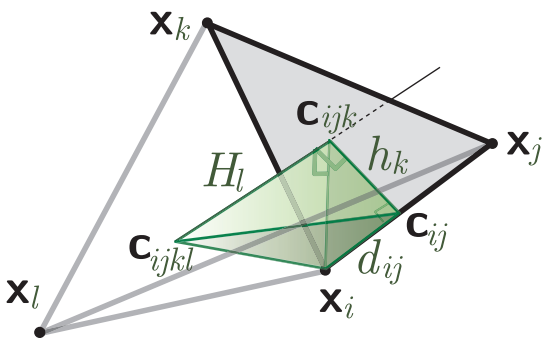
HOT Energies

B.1 $\mathrm{HOT}_{2,2}$ Energies

In this appendix, we give explicit 2-dimensional and 3-dimensional formulations of $\mathrm{HOT}_{2,2}$ energies and their gradients in a form most amenable for direct implementation. In any dimension d, $\mathrm{HOT}_{2,2}$ energies can be expressed as a function of the *signed distances* between the (weighted) circumcenters of $n-$ and $(n+1)$-simplices, for $0 \leq n \leq d-1$ (considering the primal vertices as circumcenters of 0-simplices). This is based on the simple geometric observation that the (weighted) circumcenter of an $(n+1)$-simplex projects orthogonally to the (weighted) circumcenters of its n-subsimplices. We will make use of this property when deriving closed-form expressions of the $\mathrm{HOT}_{2,2}$ energies.

Signed distances between circumcenters. Let $\mathbf{c}_{0\ldots n}$ denote the (weighted) circumcenter of simplex $[\mathbf{x}_0, \ldots, \mathbf{x}_n]$. For any $0 \leq n \leq d-1$, the signed distance between the circumcenter of the n-simplex $[\mathbf{x}_0, \ldots, \mathbf{x}_n]$ to the circumcenter of the $(n+1)$-simplex $[\mathbf{x}_0, \ldots, \mathbf{x}_{n+1}]$ is defined as the Euclidean distance between $\mathbf{c}_{0\ldots n}$ and $\mathbf{c}_{0\ldots(n+1)}$ and its sign is positive if the simplex $[\mathbf{x}_0, \ldots, \mathbf{x}_n, \mathbf{c}_{0\ldots(n+1)}]$ has the same *orientation* as $[\mathbf{x}_0, \ldots, \mathbf{x}_n, \mathbf{x}_{n+1}]$, and negative otherwise.

We further denote by d_{ij} the signed distance between the circumcenter of the 0-simplex $[\mathbf{x}_i]$ to the circumcenter of the 1-simplex $[\mathbf{x}_i, \mathbf{x}_j]$, i.e., the distance between $\mathbf{c}_i (= \mathbf{x}_i)$ and $\mathbf{c}_{ij}$, with a positive sign if $(\mathbf{x}_i - \mathbf{c}_{ij})$ has the same orientation as $(\mathbf{x}_i - \mathbf{x}_j)$, and negative otherwise. Note that d_{ji} thus corresponds to the signed distance between $\mathbf{x}_j$ and $\mathbf{c}_{ij}$. It is easy to see that $d_{ij} = \frac{|e_{ij}|^2 + w_i - w_j}{2|e_{ij}|}$, $d_{ji} = \frac{|e_{ij}|^2 + w_j - w_i}{2|e_{ij}|}$, where $e_{ij} = \mathbf{x}_j - \mathbf{x}_i$.



Going further up in dimension, we denote by h_k the signed distance between c_{ij} and c_{ijk} in a triangle

$t_{ijk} = [\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k]$. We have

$$h_k = \frac{\cot\beta_k |e_{ij}|}{2} + \frac{\cot\beta_i w_j + \cot\beta_j w_i}{2|e_{ij}|} - \frac{w_k |e_{ij}|}{4|t_{ijk}|},$$

where β_k is the angle at $\mathbf{x}_k$ in triangle t_{ijk}. Finally, we denote by H_l the signed distance between $\mathbf{c}_{ijk}$ and $\mathbf{c}_{ijkl}$ in tetrahedron T_{ijkl}.

Through the cell and simplex splitting explained in section 3.4.3, we can use the integral forms of $W(\mathbf{p}, T)$ given earlier, resulting in closed-form expressions of all the $\mathrm{HOT}_{2,2}$ energies for every triangle t_{ijk} and tetrahedron T_{ijkl} as a function of the signed distances d_{ij}, h_k, and H_l between circumcenters as follows:

$\mathbf{HOT_{2,2}}$ formulas in 2 dimensions:

$$\star^0\text{-}\,\mathrm{HOT}_{2,2}(t_{ijk}) = \sum_{i,j,k} \Big(\frac{d_{ij}^3 h_k}{4} + \frac{d_{ij} h_k^3}{12} \Big).$$

$$\star^1\text{-}\,\mathrm{HOT}_{2,2}(t_{ijk}) = \sum_{i,j,k} \Big(\frac{d_{ij}^3 h_k}{6} + \frac{d_{ij} h_k^3}{6} \Big).$$

$$\star^2\text{-}\,\mathrm{HOT}_{2,2}(t_{ijk}) = \sum_{i,j,k} \Big(\frac{d_{ij}^3 h_k}{12} + \frac{d_{ij} h_k^3}{4} \Big).$$

$\mathbf{HOT_{2,2}}$ formulas in 3 dimensions:

$$\star^0\text{-}\,\mathrm{HOT}_{2,2}(T_{ijkl}) = \sum_{i,j,k,l} \frac{1}{5} \Big(\frac{H_l^3 h_k d_{ij}}{12} + \frac{H_l h_k^3 d_{ij}}{4} + \frac{H_l h_k d_{ij}^3}{2} \Big).$$

$$\star^1\text{-}\,\mathrm{HOT}_{2,2}(T_{ijkl}) = \sum_{i,j,k,l} \frac{1}{3} \Big(\frac{H_l^3 h_k d_{ij}}{12} + \frac{H_l h_k^3 d_{ij}}{4} + \frac{H_l h_k d_{ij}^3}{6} \Big).$$

$$\star^2\text{-}\,\mathrm{HOT}_{2,2}(T_{ijkl}) = \sum_{i,j,k,l} \frac{1}{3} \Big(\frac{H_l^3 h_k d_{ij}}{6} + \frac{H_l h_k^3 d_{ij}}{4} + \frac{H_l h_k d_{ij}^3}{12} \Big).$$

$$\star^3\text{-}\,\mathrm{HOT}_{2,2}(T_{ijkl}) = \sum_{i,j,k,l} \frac{1}{5} \Big(\frac{H_l^3 h_k d_{ij}}{2} + \frac{H_l h_k^3 d_{ij}}{4} + \frac{H_l h_k d_{ij}^3}{12} \Big).$$

Vertex position optimization. In order to find the optimized position of the vertices of the mesh, we only need the derivative of the signed distances between (weighted) circumcenters with respect to vertices. Some of them may be easily derived from the formulas we provided before in this appendix. For instance:

$$\frac{\partial d_{ij}}{\partial \mathbf{x}_i} = -\frac{(\mathbf{x}_j - \mathbf{x}_i)}{2|e_{ij}|} + \frac{(w_i - w_j)(\mathbf{x}_j - \mathbf{x}_i)}{2|e_{ij}|^3}.$$

$$\frac{\partial d_{ji}}{\partial \mathbf{x}_i} = -\frac{(\mathbf{x}_j - \mathbf{x}_i)}{2|e_{ij}|} + \frac{(w_j - w_i)(\mathbf{x}_j - \mathbf{x}_i)}{2|e_{ij}|^3}.$$

More generally, we can derive all other formulas by using equation (3.7), which defines the equation for the vector from $\mathbf{x}_i$ to the (weighted) circumcenter of any simplex incident to $\mathbf{x}_i$. Through repeated uses of Pythagoras' theorem, one can then easily differentiate the signed distances between (weighted) circumcenters with respect to $\mathbf{x}_i$.

Weight optimization. The weight optimization of each $\mathrm{HOT}_{2,2}$ energy can be easily done using the following simple formulas:

$$\frac{\partial d_{ij}}{\partial w_i} = \frac{1}{2|e_{ij}|}, \quad \frac{\partial d_{ji}}{\partial w_i} = -\frac{1}{2|e_{ij}|},$$

$$\frac{\partial h_k}{\partial w_i} = \frac{\cot\theta_j}{2|e_{ij}|}, \; \frac{\partial h_j}{\partial w_i} = \frac{\cot\theta_k}{2|e_{ik}|}, \quad \frac{\partial h_i}{\partial w_i} = -\frac{|e_{jk}|}{4|t_{ijk}|}.$$

While the derivative of the weighted circumcenter with respect to primal vertices does not have a short expression, the derivative with respect to the weights can be easily computed in any dimension: this derivative at a vertex is proportional to the inverse of the distance to the opposite facet in the outward normal direction of that facet. In 2 dimensions, this results in $\partial \mathbf{c}^w_{t_{ijk}}/\partial w_i = e^{\perp}_{jk}/(4|t_{ijk}|)$, while in 3 dimensions, $\partial \mathbf{c}^w_{T_{ijkl}}/\partial w_i = \mathbf{n}_i(T_{ijkl})/(12\,|T_{ijkl}|)$, where $\mathbf{n}_i(T_{ijkl})$ is the normal (weighted by its area) of the facet opposite to $\mathbf{x}_i$ in tetrahedron T_{ijkl}.

$\star^{\mathbf{d}}$-$\mathbf{HOT_{2,2}}$-optimal weights in $\mathbb{R}^{\mathbf{d}}$: In this particular case, there is a linear expression for the optimal w_i^*'s when all other weights are considered fixed. In 2 dimensions, we get

$$w_i^* = \frac{2\sum_{j\in\Omega(i)}\Big(\cot(\beta_k)+\cot(\beta_l)\Big)w_j + 4\sum_{t_{ijk}}\big(\mathbf{c}_t - \mathbf{b}_t\big)\cdot e^{\perp}_{jk}}{\sum_{t_{ijk}}\frac{\|e_{jk}\|^2}{|t_{ijk}|}}$$

where β_k is the angle at $\mathbf{x}_k$ in triangle t_{ijk}, and $\mathbf{b}_t$ is the barycenter of the triangle t_{ijk}. In 3 dimensions, we have instead:

$$w_i^* = \Bigg(\sum_{T_{ijkl}\in\Omega(i)}\Big[w_j\cot(\alpha_{kl})|e_{kl}| + w_k\cot(\alpha_{jl})|e_{jl}| + w_l\cot(\alpha_{jk})|e_{jk}| + 2(\mathbf{c}_T - \mathbf{b}_T)\cdot\mathbf{n}_i(T_{ijkl})\Big]\Bigg) \Big/ \Bigg(\sum_{T_{ijkl}\in\Omega(i)}\frac{2\,|t_{jkl}|^2}{3\,|T_{ijkl}|}\Bigg)$$

where α_{kl} is the dihedral angle at edge e_{kl}.

B.2 $\mathrm{HOT}_{1,1}$ Energies

In this appendix, we give explicit 2-dimensional and 3-dimensional formulations of $\mathrm{HOT}_{1,1}$ energies for those which are practical to compute.

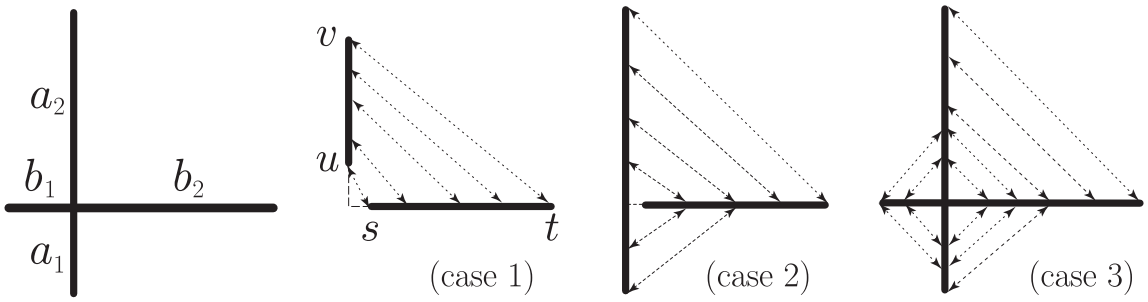


Figure B.1. $\mathbf{W_1}$ *transport between orthogonal edges*: *For two arbitrary edges parameterized by the length a_1, a_2, b_1, and b_2 (left), the transport plan (sampled in dotted lines) and cost of the normalized measure from one edge to the other can be computed in closed form for the three possible configurations displayed above. Notice that the measure of an edge is sometimes evenly transported to two locations on the other edge (cases 2 and 3).*

$\mathbf{HOT_{1,1}}$ for $\star^{\mathbf{0}}$ and $\star^{\mathbf{d}}$ in $\mathbb{R}^{\mathbf{2}}$ and $\mathbb{R}^{\mathbf{3}}$. Using the cell and simplex splitting used in section 3.4.3, we can derive closed-form expressions for both $\star^0$-$\mathrm{HOT}_{1,1}(\mathcal{M})$ and $\star^d$-$\mathrm{HOT}_{1,1}(\mathcal{M})$ as these two cases only involve transport plans between convex regions and points. In 2 dimensions we can express the integral over a right triangle with width $a > 0$ and height $b > 0$ to the point $\mathbf{p}$ adjacent to a (see figure 3.5(left)) as

$$\begin{aligned} W(\mathbf{p}, T) &= \int_0^a \int_0^{\frac{bx}{a}} \sqrt{x^2+y^2}\, dy\, dx \\ &= \frac{1}{6}\left[ab\sqrt{a^2+b^2} + a^3 \sinh^{-1}\left(\frac{b}{a}\right)\right]. \end{aligned}$$

The gradients of these energies are then expressed as

$$\frac{\partial W(\mathbf{p}, T)}{\partial a} = \frac{1}{6}\left[b\sqrt{a^2+b^2} + 3a^2 \sinh^{-1}\left(\frac{b}{a}\right)\right],$$

and

$$\frac{\partial W(\mathbf{p}, T)}{\partial b} = \frac{1}{3} a\sqrt{a^2+b^2}.$$

The 3-dimensional counterpart for a biorthogonal tetrahedron and one of its vertices (see figure 3.5 (right)) becomes:

$$\begin{aligned} W(\mathbf{p}, T) = &\int_0^a \int_0^{\frac{bx}{a}} \int_0^{\frac{cy}{b}} \sqrt{x^2+y^2+z^2}\, dz dy dx \\ &= \frac{1}{24}\Big(abc\sqrt{a^2+b^2+c^2} + ab\left(b^2+3a^2\right)\sinh^{-1}\Big(\frac{c}{\sqrt{a^2+b^2}}\Big) \\ &\quad + 3a^4\Big[\tan^{-1}\Big(\frac{b}{c}\Big) - \tan^{-1}\Big(\frac{l\sqrt{a^2+b^2+c^2}}{ac}\Big)\Big] \\ &\quad + a^4\Big[\tan^{-1}\Big(\frac{c}{b}\Big) - \tan^{-1}\Big(\frac{ac}{l\sqrt{a^2+b^2+c^2}}\Big)\Big]\Big). \end{aligned}$$

The algorithm in figure 3.3 can then be applied directly using these expressions. As expected for $\star^0$, we obtain what could be called a "Medial Voronoi Tesselation": each vertex is on (or very near) the geometric median of its weighted Voronoi regions, see figure 3.10. (Note that when the weights are

all constant, a Lloyd-like algorithm could also be used, for which each vertex is repeatedly moved to the median of its own Voronoi cell; we found, however, the quasi-Newton method with Wolfe conditions to be more efficient as in the $\mathrm{HOT}_{2,2}$ case.) Similarly for $\star^d$, the resulting meshes have their dual complex centered at the median of each triangle.

$\star^1$-$\mathbf{HOT}_{1,1}$ in 2 dimensions. The optimal transport cost from a primal edge to a dual edge for the 1-Wasserstein distance can be computed in closed form with some effort. We first compute the optimal transport cost between two separated, orthogonal edges (depicted in figure B.1) parameterized as

$$W(s,t,u,v) = \int_s^t \sqrt{x^2 + \left(\frac{v-u}{t-s}(x-s)+u\right)^2}\,dx, \tag{B.1}$$

where $t > s \geq 0$ and $v > u \geq 0$, for which a closed-form expression can be found in Mathematica. The general cost E is then computed as one of three possible edge configurations depicted in figure B.1 (with $a = a_1 + a_2, b = b_1 + b_2$):

- $b_1 < 0$ and $a_1 < 0$: (case 1)

  $E = \frac{a}{2}\, W(|b_1|, b_2, |a_1|, a_2)$

- $b_1 < 0$ and $a_1 > 0$: (case 2)

  $E = \frac{a}{2}(W(|b_1|, 2a_1b/a + |b_1|, 0, a_1) + W(2a_1b/a + |b_1|, b_2, a_1, a_2))$

- Otherwise: (case 3)

  $E = \frac{a}{2}(2W(0, b_1, 0, ab_1/b) + W(d_1, 2a_1b/a - b_1, ab_1/b, a_1)$
  $\quad + W(2a_1b/a - b_1, b_2, a_1, a_2))$.

The optimality of these transport plans can be seen by noting that, by the triangle inequality of the Euclidean metric, transport lines for the optimal transport plans cannot cross (see chapter 8 of [70]). This uniquely defines the optimal transport plan for case 1, while the density splitting in cases 2 and 3 follow from this and the symmetry of the regions in which the density is split.

Appendix C

Finite Volume Discretization of Advection Term

We briefly provide a derivation of the advection term in this section. With first-order accuracy, we rewrite the area integral involved as

$$\begin{aligned}\int_{f_{ij}} (\boldsymbol{\omega}\times\mathbf{u})\cdot\mathbf{n}\,dA &\simeq -\int_{f_{ij}} \nabla\times(\mathbf{u}\cdot(x-c_{ij})\boldsymbol{\omega})\cdot\mathbf{n}\,dA \\ &= -\int_{\partial f_{ij}} (\mathbf{u}\cdot(x-c_{ij}))\boldsymbol{\omega}\cdot\,\mathbf{dl}.\end{aligned}$$

Consequently, $\mathrm{Adv}(F)_{ij}$ can be approximated by summing, over the edges of face f_{ij}, the product of $\boldsymbol{\omega}$ dotted with an edge e and $\mathbf{u}$ dotted with $h_{e,ij} = c_{ij} - c_e$ (midpoint quadrature). Thus, using the vorticity ω_e approximated on face s_e dual to edge e, the contribution from flux F_{jk} to $\dot{F}_{ij}$ is

$$\frac{\omega_e|e|}{|s_e|}(\mathbf{u}\cdot h_{e,ij}) = \frac{\omega_e|e|}{|s_e|\sin(\alpha)}\left(\frac{F_{jk}}{|f_{jk}|} - \cos(\alpha)\frac{F_{ij}}{|f_{ij}|}\right)|h_{e,ij}|,$$

where α is the dihedral angle of tet T_j at edge e: indeed, $\mathbf{u}$ projected onto the direction of $h_{e,ij}$ is determined by the projection of $\mathbf{u}$ onto the plane orthogonal to e, i.e., by its projections $F_{ij}/|f_{ij}|$ and $F_{jk}/|f_{jk}|$ on two noncolinear directions within that plane. We then convert the above expression to $\star_2\mathrm{Adv}(F)$ by multiplying through $|h_{ij,j}|/|f_{ij}|$:

$$\frac{\boldsymbol{\omega}_e}{|s_e|}\left(F_{jk}\frac{4|f_{ij,e}||h_{ij,j}|}{3|T_j||e|} - F_{ij}\frac{2|f_{ij,e}||h_{ij,j}|}{|f_{ij}|^2\cot\alpha}\right),$$

where $f_{ij,e}$ is the triangle formed by e and the circumcenter c_{ij} of the face, and $h_{ij,j}$ is the partial dual edge in cell j.

Notice that in the continuous limit, one should have $\int(\boldsymbol{\omega}\times\mathbf{u})\times\mathbf{u} = 0$ for arbitrary $\boldsymbol{\omega}$ and $\mathbf{u}$. We enforce this property at the discrete level by further ensuring that its discrete equivalent, $F\star_2\mathrm{Adv}(F,\omega) = 0$, is automatically satisfied. It is easy to see that if the discrete version of $\int(\boldsymbol{\omega}\times\mathbf{u})\times\mathbf{u} = 0$

is to hold for all $\boldsymbol{\omega}$ and $\mathbf{u}$, we must *antisymmetrize* the contribution from F_{ij} to $\star_2\mathrm{Adv}(F)_{jk}$ and from F_{jk} to $\star_2\mathrm{Adv}(F)_{ij}$. Thus, the contribution of $\boldsymbol{\omega}_e$ and F_{jk} to $\star_2\mathrm{Adv}(F)_{ij}$ is rectified to

$$2\frac{\boldsymbol{\omega}_e}{|s_e|}F_{jk}\frac{|f_{ij,e}||h_{ij,j}|+|f_{ik,e}||h_{ik,j}|}{3|T_j||e|}=\frac{\boldsymbol{\omega}_e}{|s_e|}F_{jk}\frac{2|w_{e,j}|}{3|T_j|},$$

where $w_{e,j}$ is the quad-shaped partial dual face of e in cell j.

This discretization of the advection term now enforces energy preservation. Indeed, if we denote $\bar{F}^{n+\frac{1}{2}}:=(F^n+F^{n+1})/2$ as before, our antisymmetrization of the finite-volume approximation of the change of flux directly yields

$$\bar{F}^{n+\frac{1}{2}}\star_2\mathrm{Adv}(\bar{F}^{n+\frac{1}{2}})=0.$$

Multiplying the time update in equation (4.5) by $\bar{F}^{n+\frac{1}{2}}$ proves that this condition enforces that the energy at time t^{n+1} is equal to the energy at time t^n.

Bibliography

[1] J. J. Monaghan, "Smoothed particle hydrodynamics," *Reports on Progress in Physics*, vol. 68, no. 8, p. 1703, 2005.

[2] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '87. New York, NY, USA: ACM, 1987, pp. 205–214.

[3] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, 2nd ed., ser. Monographs on Appl. Comput. Math. Cambridge: Cambridge University Press, 1999, vol. 3.

[4] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, ser. Applied Mathematical Sciences. New York: Springer-Verlag, 2003, vol. 153.

[5] C. Hirt and B. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *Journal of Computational Physics*, vol. 39, no. 1, pp. 201–225, 1981.

[6] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, "A hybrid particle level set method for improved interface capturing," *J. Comput. Phys*, vol. 183, pp. 83–116, 2002.

[7] J. Donéa and A. Huerta, *Finite Element Methods for Flow Problems*. Wiley, 2003.

[8] D. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, and M. Shashkov, Eds., *Compatible Spatial Discretizations*, ser. I.M.A. Volumes. Springer, 2006, vol. 142.

[9] A. Bossavit, *Computational Electromagnetism*. Boston: Academic Press, 1998.

[10] A. Stern, Y. Tong, M. Desbrun, and J. E. Marsden, "Variational integrators for Maxwell's equations with sources," in *Progress in Electromagnetics Research Symposium*, vol. 4, no. 7, June 2008, pp. 711–715.

[11] R. A. Nicolaides and X. Wu, "Covolume solutions of three dimensional div-curl equations," *SIAM J. Numer. Anal.*, vol. 34, p. 2195, 1997.

[12] P. B. Bochev and J. M. Hyman, "Principles of mimetic discretizations of differential operators," *I.M.A. Volumes*, vol. 142, pp. 89–119, 2006.

[13] A. N. Hirani, "Discrete exterior calculus," PhD dissertation, California Institute of Technology, May 2003.

[14] M. Desbrun, E. Kanso, and Y. Tong, "Discrete differential forms for computational sciences," in *Discrete Differential Geometry*, ser. Course Notes, E. Grinspun, P. Schröder, and M. Desbrun, Eds. ACM SIGGRAPH, 2006.

[15] D. N. Arnold, R. S. Falk, and R. Winther, "Finite element exterior calculus, homological techniques, and applications," *Acta Numerica*, vol. 15, pp. 1–155, 2006.

[16] É. Cartan, *Les Systèmes Differentiels Exterieurs et leurs Applications Géometriques.* Paris: Hermann, 1945.

[17] J. R. Munkres, *Elements of Algebraic Topology.* Menlo Park, CA: Addison-Wesley, 1984.

[18] W. L. Burke, *Applied Differential Geometry.* Cambridge University Press, 1985.

[19] R. Abraham, J. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications.* Applied Mathematical Sciences Vol. 75, Springer, 1988.

[20] D. Lovelock and H. Rund, *Tensors, Differential Forms, and Variational Principles.* Dover Publications, 1993.

[21] H. Flanders, *Differential Forms and Applications to Physical Sciences.* Dover Publications, 1990.

[22] S. Morita, *Geometry of Differential Forms*, ser. Translations of Mathematical Monographs, vol. 201. Am. Math. Soc., 2001.

[23] S. Carroll, *Spacetime and Geometry: An Introduction to General Relativity.* Pearson Education, 2003.

[24] T. Frankel, *The Geometry of Physics*, ser. Second Edition. United Kingdom: Cambridge University Press, 2004.

[25] H. Whitney, *Geometric Integration Theory.* Princeton: Princeton Press, 1957.

[26] J. Nédélec, "Mixed Finite Elements in 3D in H(div) and H(curl)," *Springer Lectures Notes in Mathematics*, vol. 1192, 1986.

[27] D. N. Arnold, R. S. Falk, and R. Winther, "Finite element exterior calculus: From Hodge theory to numerical stability," *Bull. Amer. Math. Soc.*, vol. 47, no. 2, pp. 281—354, 2010.

[28] S. Elcott, Y. Tong, E. Kanso, P. Schröder, and M. Desbrun, "Stable, circulation-preserving, simplicial fluids," *ACM Trans. on Graphics*, vol. 26, no. 1, p. 4, Jan 2007.

[29] X. Gu and S.-T. Yau, "Global conformal surface parameterization," in *Symposium Geometry Processing*, 2003, pp. 127–137.

[30] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun, "Designing quadrangulations with discrete harmonic forms," in *Proc. Symp. Geometry Processing*, 2006, pp. 201–210.

[31] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for ODEs*. Springer, 2002.

[32] R. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, ser. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[33] J. Shi, C. Hu, and C. Shu, "A technique for treating negative weights in WENO schemes," *J. Comput. Phys.*, vol. 175, pp. 108–127, 2002.

[34] T. Dupont and Y. Liu, "Back-and-forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function," *Journal of Computational Physics*, vol. 190, no. 1, pp. 311–324, 2003.

[35] V. A. Titarev and E. F. Toro, "Finite-volume WENO schemes for three-dimensional conservation laws," *J. Comput. Phys.*, vol. 201, no. 1, pp. 238–260, 2004.

[36] A. Iske and M. Käser, "Conservative semi-Lagrangian advection on adaptive unstructured meshes," *Numerical Methods for Partial Differential Equations*, vol. 20, no. 3, pp. 388–411, 2004.

[37] D. J. Hill and D. I. Pullin, "Hybrid Ttned center-difference-WENO method for large eddy simulations in the presence of strong shocks," *J. Comput. Phys.*, vol. 194, no. 2, pp. 435–450, 2004.

[38] V. Dyadechko and M. Shashkov, "Moment-of-fluid interface reconstruction," LANL Technical Report LA-UR-05-7571, 2006.

[39] B. Engquist and S. Osher, "One-sided difference schemes and transonic flow," *PNAS*, vol. 77, no. 6, pp. 3071–3074, 1980.

[40] C. Shu and S. Osher, "Efficient implementation of essentially non-oscillatory shock capturing schemes," *J. Sci. Comput.*, vol. 77, pp. 439–471, 1988.

[41] X. Liu, S. Osher, and T. Chan, "Weighted essentially non-oscillatory schemes," *J. Sci. Comput.*, vol. 126, pp. 202–212, 1996.

[42] A. Bossavit, "Extrusion, contraction: Their discretization via Whitney forms." *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 22, no. 3, pp. 470–480, 2003.

[43] H. Heumann and R. Hiptmair, "Extrusion contraction upwind schemes for convection-diffusion problems," ETH Zürich, Seminar für Angewandte Mathematik SAM 2008-30, October 2008.

[44] R. J. Leveque, "CLAWPACK, at http://www.clawpack.org," 1994—2009.

[45] F. H. Harlow and J. E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *Physics of Fluids*, vol. 8, no. 12, pp. 2182–2189, Dec. 1965.

[46] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, ser. Lecture Notes in Mathematics. Springer, 1998, vol. 1697, pp. 325–432.

[47] Y. Zhang and C. Shu, "High-order WENO schemes for Hamilton-Jacobi equations on triangular meshes," *J. Sci. Comput.*, vol. 24, pp. 1005–1030, 2003.

[48] D. Levy, S. Nayak, C. Shu, and Y. Zhang, "Central WENO schemes for Hamilton-Jacobi equations on triangular meshes," *J. Sci. Comput.*, vol. 27, pp. 532–552, 2005.

[49] J. Shewchuk, "What is a good linear element? Interpolation, conditioning, and quality measures," in *Proc. of the 11th Int. Meshing Roundtable*, 2002, pp. 115–126.

[50] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, pp. 637–676, December 1999.

[51] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun, "Variational tetrahedral meshing," *ACM Trans. on Graphics (SIGGRAPH)*, vol. 24, no. 3, pp. 617–625, July 2005.

[52] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer-Verlag, 2003, pp. 35–57.

[53] C. Batty, S. Xenos, and B. Houston, "Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids," *Computer Graphics Forum (Eurographics)*, vol. 29, pp. 695–704(10), May 2010.

[54] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*. Springer-Verlag, 1987.

[55] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed., ser. Probability and Statistics. Wiley, 2000.

[56] M. Desbrun, E. Kanso, and Y. Tong, "Discrete differential forms for computational modeling," in *Discrete Differential Geometry*, A. Bobenko and P. Schröder, Eds. Springer, 2007.

[57] L. J. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, 2010.

[58] J. B. Perot and V. Subramanian, "Discrete calculus methods for diffusion," *Journal of Computational Physics*, vol. 224, pp. 59–81, 2007.

[59] J. Warren, S. Schaefer, A. Hirani, and M. Desbrun, "Barycentric coordinates for convex sets," *Advances in Computational Mathematics*, vol. 27, no. 3, pp. 319–338, 2007.

[60] V. Rajan, "Optimality of the Delaunay triangulation in $\mathbb{R}^d$," *Discrete and Computational Geometry*, vol. 12, no. 1, pp. 189–202, 1994.

[61] E. VanderZee, A. N. Hirani, D. Guoy, and E. Ramos, "Well-centered triangulation," *SIAM Journal on Scientific Computing*, vol. 31, no. 6, pp. 4497–4523, 2010.

[62] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental Mathematics*, vol. 2(1), pp. 15–36, 1993.

[63] A. Bossavit, *Computational Electromagnetism*. BOSTON: Academic Press.

[64] B. Auchmann and S. Kurz, "A geometrically defined discrete Hodge operator on simplicial cells," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 643–646, 2006.

[65] S. O. Wilson, "Cochain algebra on manifolds and convergence under refinement," *Topology and its Applications*, vol. 154, no. 9, pp. 1898 – 1920, 2007.

[66] M. Fisher, B. Springborn, A. I. Bobenko, and P. Schröder, "An algorithm for the construction of intrinsic delaunay triangulations with applications to digital geometry processing," in *ACM SIGGRAPH Courses*, 2006, pp. 69–74.

[67] J. R. Munkres, *Elements of Algebraic Topology*. Addison-Wesley, 1984.

[68] D. Glickenstein, "Geometric triangulations and discrete Laplacians on manifolds," *Arxiv preprint math/0508188*, 2005.

[69] D. Pedoe, *Geometry, a Comprehensive Course*, 2nd ed. Dover Publications, 1988.

[70] C. Villani, *Optimal Transport: Old and New*, ser. Fundamental Principles of Mathematical Sciences, 338. Springer-Verlag, 2009.

[71] F. Mémoli, "A spectral notion of Gromov-Wasserstein distances and related methods," *Journal of Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 363–401, 2011.

[72] Y. Lipman and I. Daubechies, "Surface comparison with mass transportation," in *ArXiv preprint 0912.3488*, 2010.

[73] P. Hauret, E. Kuhl, and M. Ortiz, "Diamond elements: A finite element/discrete-mechanics approximation scheme with guaranteed optimal convergence in incompressible elasticity," *Int. J. Numer. Meth. Engng.*, vol. 72, no. 3, pp. 253–294, 2007.

[74] Y. Liu, W. Wang, B. Lévy, F. Sun, D. Yan, L. Lu, and C. Yang, "On centroidal Voronoi tessellation—Energy smoothness and fast computation," *ACM Trans. on Graph.*, vol. 28, no. 4, 2009.

[75] B. Lévy and Y. Liu, "L_p centroidal Voronoi tesselation and its applications," *ACM Trans. on Graph.*, vol. 29, no. 4, 2010.

[76] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Verlag, 1999.

[77] CGAL, "Computational Geometry Algorithms Library (release 3.8)," 2010, http://www.cgal.org.

[78] J. Tournois, C. Wormser, P. Alliez, and M. Desbrun, "Interleaving Delaunay refinement and optimization for practical isotropic tetrahedron mesh generation," *ACM Trans. Graph.*, vol. 28, pp. 75:1–75:9, July 2009.

[79] D. Sieger, P. Alliez, and M. Botsch, "Optimizing Voronoi diagrams for polygonal finite element computations," in *Proceedings of the 19th International Meshing Roundtable*, 2010, pp. 335–350.

[80] S.-W. Cheng, T. K. Dey, and J. Levine, "Theory of a practical Delaunay meshing algorithm for a large class of domains," in *Algorithms, Architecture and Information Systems Security*, ser. World Scientific Review, B. Bhattacharya, S. Sur-Kolay, S. Nandy, and A. Bagchi, Eds., vol. 3, 2008, pp. 17–41.

[81] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, "Discrete Laplace operators: No free lunch," in *Symposium on Geometry processing*, 2007, pp. 33–37.

[82] N. Foster and D. Metaxas, "Modeling the motion of a hot, turbulent gas," in *Proceedings of SIGGRAPH*, Aug. 1997, pp. 181–188.

[83] R. Fedkiw, J. Stam, and H. W. Jensen, "Visual simulation of smoke," in *Proceedings of ACM SIGGRAPH*, 2001, pp. 15–22.

[84] J. Stam, "Stable fluids," in *Proceedings of ACM SIGGRAPH*, Aug. 1999, pp. 121–128.

[85] J. Steinhoff and D. Underhill, "Modification of the Euler equations for *vorticity confinement*," *Physics of Fluids*, vol. 6, no. 8, pp. 2738–2744, Aug. 1994.

[86] B. Kim, Y. Liu, I. Llamas, and J. Rossignac, "Advections with significantly reduced dissipation and diffusion," *IEEE Trans. on Visualiz.and Comp. Graphics*, vol. 13(1), pp. 135–144, 2007.

[87] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac, "An unconditionally stable MacCormack method." *J. Sci. Comp.*, vol. 35, pp. 350–371, 2008.

[88] J. Molemaker, J. M. Cohen, S. Patel, and J. yong Noh, "Low viscosity flow simulations for animation," in *Symposium on Computer Animation*, Jun 2008, pp. 9–18.

[89] H. Schechter and R. Bridson, "Evolving sub-grid turbulence for smoke animation," in *Symposium on Computer Animation*, Jun 2008, pp. 1–8.

[90] A. Selle, N. Rasmussen, and R. Fedkiw, "A vortex particle method for smoke, water and explosions," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 910–914, Aug. 2005.

[91] Y. Zhu and R. Bridson, "Animating sand as a fluid," in *Proceedings of ACM SIGGRAPH*, 2005, pp. 965–972.

[92] J. Brackbill and H. Ruppel, "FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions," *Journal of Computational Physics*, vol. 65, pp. 314–343, 1986.

[93] L. Shi and Y. Yu, "Visual smoke simulation with adaptive octree refinement," *Computer Graphics and Imaging*, 2002.

[94] F. Losasso, F. Gibou, and R. Fedkiw, "Simulating water and smoke with an octree data structure," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 457–462, Aug. 2004.

[95] B. E. Feldman, J. F. O'Brien, and B. M. Klingner, "Animating gases with hybrid meshes," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 904–909, 2005.

[96] N. Chentanez, B. E. Feldman, F. Labelle, J. F. O'Brien, and J. Shewchuk, "Liquid simulation on lattice-based tetrahedral meshes," in *Symposium on Computer Animation*, Aug. 2007, pp. 219–228.

[97] S. Elcott, Y. Tong, E. Kanso, P. Schröder, and M. Desbrun, "Stable, circulation-preserving, simplicial fluids," *ACM Transactions on Graphics*, vol. 26, no. 1, p. art. 4, Jan. 2007.

[98] B. Perot, "Conservation properties of unstructured staggered mesh schemes," *J. Comput. Phys.*, vol. 159, no. 1, pp. 58–89, 2000.

[99] X. Zhang, D. Schmidt, and B. Perot, "Accuracy and conservation properties of a 3D unstructured staggered mesh scheme for fluid dynamics," *J. Comput. Phys.*, vol. 175, no. 2, pp. 764–791, 2002.

[100] K. Mahesh, G. Constantinescu, and P. Moin, "A numerical method for large-eddy simulation in complex geometries," *J. Comput. Phys.*, vol. 197, no. 1, pp. 215–240, 2004.

[101] A. Stern and M. Desbrun, "Discrete geometric mechanics for variational time integrators," in *ACM SIGGRAPH Course Notes*, 2006, pp. 75–80.

[102] A. Chorin and J. Marsden, *A Mathematical Introduction to Fluid Mechanics*, 3rd ed. Springer-Verlag, 1979.

[103] P. M. Gresho and R. L. Sani, *Incompressible Flow and the Finite Element Method.* J. Wiley & Sons, 2000.

[104] M. Duponcheel, P. Orlandi, and G. Winckelmans, "Time-reversibility of the Euler equations as a benchmark for energy conserving schemes," *Journal of Computational Physics*, vol. 227, no. 19, pp. 8736–8752, 2008.

[105] B. E. Feldman, J. F. O'Brien, and O. Arikan, "Animating suspended particle explosions," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 708–715, July 2003.

[106] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, pp. 1–137, 2005.

[107] J. Simo and F. Armero, "Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier-Stokes and Euler equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 111, no. 1-2, pp. 111–154, 1994.

[108] D. Pavlov, "Structure-preserving discretizations of incompressible fluids," PhD dissertation, California Institute of Technology, 2009.

[109] "Geometric, variational discretization of continuum theories," *Physica D: Nonlinear Phenomena*, vol. In Press.

[110] M. Bergou, S. Mathur, M. Wardetzky, and E. Grinspun, "TRACKS: Toward directable thin shells," *ACM Trans. on Graphics*, vol. 26, no. 3, p. art. 50, 2007.

[111] T. Regge, "General relativity without coordinates," *Il Nuovo Cimento (1955-1965)*, vol. 19, pp. 558–571, 1961, 10.1007/BF02733251.

[112] M. Bouman, A. Palha, J. Kreeft, and M. Gerritsma, "A conservative spectral element method for curvilinear domains," in *Spectral and High Order Methods for Partial Differential Equations*, ser. Lecture Notes in Computational Science and Engineering, T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, T. Schlick, J. S. Hesthaven, and E. M. Rnquist, Eds. Springer Berlin Heidelberg, 2011, vol. 76, pp. 111–119, 10.1007/978-3-642-15337-2_8.

[113] N. Robidoux, "Polynomial histopolation, superconvergent degrees of freedom, and pseudospectral discrete Hodge operators," preprint available at http://www.cs.laurentian.ca/robidoux/.

[114] G.-S. Jiang and C. Shu, "Efficient implementation of weighted ENO schemes," *J. Comput. Phys.*, vol. 126, pp. 202–228, 1996.